Spiro AI:
Smartphone Based Pulmonary Function Testing

Jake Garrison

A thesis

submitted in partial fulfillment of the

requirements for the degree of

Master of Science in Electrical Engineering

University of Washington

2018

Committee:

Shwetak Patel

Robert Bruce Darling

Program Authorized to Offer Degree:

Electrical Engineering

University of Washington

Abstract

Spiro AI: Smartphone Based Pulmonary Function Testing

Jake Garrison

Chair of the Supervisory Committee:

Shwetak Patel

Department of Electrical Engineering

Spirometry is a widely employed pulmonary function test used to benchmark lung health and assist in diagnosing chronic lung conditions such as chronic obstructive pulmonary disease and asthma. When used frequently, such as in a home or portable setting, spirometry results can predict pulmonary exacerbations or monitor the effectiveness of treatment. Unfortunately, portable options are expensive and not truly portable by modern standards. Prior work has shown it is possible to conveniently obtain spirometry metrics using the built-in microphone of a smartphone, requiring no accessories. This work proposes Spiro AI, an end to end sound-based smartphone spirometry system that includes automatic quality control and complete spirometry reporting, bringing smartphone spirometry closer to reality. Several machine learning models and deep learning architectures are thoroughly evaluated as potential components in the system. Models are trained and evaluated on thousands of patients sourced from a newly created dataset that is likely the largest audio based spirometry dataset to date. The results suggest the problem becomes increasingly difficult when the

sample size scales from tens to thousands of subjects because the population is more diverse and the quality of recorded maneuvers becomes difficult to control. Nonetheless, the results suggest Spiro AI is capable of trend reporting and screening; however, in its current stage it may not be precise enough for FDA certification.

Contents

1 Introduction

1.1 Overview

There are over two billion smartphone users in the world who spend an average of two hours per day checking, searching, replying and browsing various applications, yet only average one phone call per day. The smartphone is well beyond being considered a smarter version of a phone; to many, it is a replacement to a music player, camera, notebook, calculator and even computer. In fact, more people in the world have access to smartphones than working toilets and because of smartphones, the line for these toilets has never been longer. Smartphones, or rather, smart-appendages tell us everything we need to know whenever we want to know it similar to a prompt personal assistant, or divine, all-knowing oracle.

Smartphones are equipped with several sensors to facilitate native functionality such as sound and image capture, navigation, screen rotation and touch input to name a few. These sensors can also be exploited for supplemental purposes and integrated to enable new, innovative uses. For example, activity or fitness tracking applications fuse motion and location information to discern the difference between running, biking, driving or sleeping. This idea of exploiting something ubiquitous to conveniently serve another purpose is not a new concept. Our prehistoric ancestors discovered that hollowing out a tree makes water commuting more convenient and sharpening a rock makes hunting more productive. It is our innate ability to make lemonade from lemons that has transformed portable phones into the digital swiss army knife that dominates the lives of a quarter of the world's population.

One innovative use of smartphones *mobile health* involves a niche subset of mobile applications that leverage embedded sensors in novel ways to measure and monitor information relevant to an individual's health and wellness. Mobile health applications typically measure a relevant signal, such as motion or pulse, and sometimes use this to inform more generalized insights such as sleep quality, calories burned or stress level. These signals are often obtained through an alternative use of one or more sensors. For example, steps can be detected and counted through observing a specific pattern of motion and even pulse can be inferred from sensing the variation of color in a vein using a camera. Many of these informative signals have trusted but less convenient measurement techniques, such as a heart rate monitor. Rather than explicitly defining the patterns or variations that allow these signals to be

measured on a smartphone, examples can be accumulated from both the smartphone and trusted technique. These examples can then be used to teach a computer to automatically interpret the desired signal from the original sensor information. The idea of teaching a computer to learn through example is known as *machine learning* and is also implicitly a form of *artificial intelligence* as it involves an artificial entity applying learned knowledge to perceive information in an adaptive environment. With machine learning, our computers make the lemonade for us.

This work described in this thesis is focused on the mobile health application of *spirometry*, which is a common measurement technique for assessing lung function. This work, titled *Spiro AI*, utilizes the sound of an individual's exhale recorded via smartphone microphone along with various machine learning strategies to report metrics common to spirometry. These metrics, if accurate, can be used to screen for various respiratory conditions or track symptoms and treatment effectiveness for those already diagnosed. Furthermore, performing spirometry on a smartphone is far more convenient and affordable for the patient and provides the care providers with a more detailed lung function assessment which can, in turn, lead to better health outcomes for the patient.

1.1.1 Motivation

Chronic obstructive pulmonary disease (COPD) is currently the third leading cause of death in the world and unlike many of the other top causes which have stabilized or even decreased in prevalence, COPD is taking lives at an accelerating rate. Other potentially fatal respiratory diseases such as lung cancer and tuberculosis are also listed as top causes of death. Together, these diseases account for one-sixth of all global deaths and by 2030, they are estimated to contribute to one-fifth of all deaths due to the accelerating mortality rate of COPD, despite the decreasing rate of lung cancer and tuberculosis. Unfortunately, COPD has no cure and by the time COPD is typically diagnosed the damage cannot be reversed. The symptoms, however, can be managed using various treatment options. Prior to diagnosis, the progression can be slowed or minimized if exposure to risk factors such as smoking or air pollution is reduced.

Spirometry is the measurement tool used by physicians to assess lung function in order to screen for at-risk patients or to evaluate treatment options for those already diagnosed with COPD or a number of other respiratory conditions such as asthma and cystic fibrosis. Spirometry is typically conducted in the physician's office as it requires a specific maneuver to be performed which often requires coaching to ensure the effort is completed correctly. There are also expensive home spirometry options as well as limited, portable variants. For those who benefit from tracking the progression of a lung condition or treatment effectiveness, the usefulness of spirometry is determined by the frequency at which measurements are recorded. Clearly, there exists a dire need for affordable, portable spirometry options,

especially for patients at risk or already diagnosed with conditions such as COPD, asthma and cystic fibrosis. This work explores the prospect of performing spirometry tests with only a smartphone as a potential solution to many of the issues surrounding screening and management of respiratory diseases.

Spiro AI is based on the pioneering SpiroSmart publication which first proposed and evaluated a smartphone-based solution to spirometry six years ago [46]. Since the advent of SpiroSmart, collaborations have formed between the University of Washington, Seattle Children's Hospital and clinics around the world with the goal of creating a massive dataset for validating and improving the algorithms that power SpiroSmart. Today, there exists smartphone audio recordings and spirometry ground truth for over 4000 different patients. Additionally, several critical advancements in the field of artificial intelligence have surfaced in the last decade which enables far more powerful predictive algorithms to be developed using large amounts of data. This work takes advantage of this new data, as well as recent advancements in artificial intelligence to propose and evaluate machine learning based methods for smartphone-based spirometry.

1.1.2 Problem Statement

The purpose of this work is less about offering a single solution to smartphone spirometry and more about investigating various machine learning strategies in order to understand which techniques are most effective and practical as potential solutions. Therefore, the problem statement is:

Explore data-driven methods for computing spirometry metrics suitable for respiratory disease management and screening from a smartphone sound recording of a forced expiratory maneuver.

1.2 Results

The methods explored in this work are evaluated on thousands of different trials from hundreds of different patients with various lung conditions whereas prior work evaluated proposed methods on a sample size of around 50 local, mostly healthy subjects. The effectiveness of a particular method is measured based on the overall error of various spirometry metrics, as well the efficiency and feasibility of employing the method on a smartphone. The Spiro AI system is comprised of several subsystems, each with a separate evaluation strategy. The final error results are listed for each subsystem along with accompanying plots and insights.

Overall, the results suggest Spiro AI is indeed a promising and complete end to end smartphone-based solution that will need further validation with both longitudinal studies, as well as more diverse latitudinal studies before it can be considered a solution mature enough for regulatory, Food and Drug Administration (FDA) certification.

1.2.1 Contributions

The main technical contributions derived from this work as a whole are listed below in relative order of magnitude:

- *Spiro AI*, an end to end smartphone spirometry testing system
- The largest known dataset of cleansed and organized spirometry audio recordings with paired, reproducible ground truth
- *CurveNet*, a novel sound to airflow neural network architecture bounded by physics
- A production-ready backend and iOS app suitable for Spiro AI demonstrations and future data collection and user studies
- A generalized, extendable preprocessing, training and evaluation pipeline for use in other audio-based machine learning problems
- Open source ultrasonic sensing and DIY spirometry toolkits

A less technical, but equally imperative contribution lies in the structure of this thesis. The importance of multidisciplinary collaboration is stressed throughout this work and it would, therefore, be hypocritical to only focus on the technical contributions without exploring the background for *why* it is important and *how* it all works in a manner that is informative for a general audience. Mobile health by definition requires expertise and input from engineers, physicians, care providers, and regulators. A true mobile health solution will only arise if these disciplines are speaking the same language and working together from the same foundation.

1.3 Thesis Structure

Since one of the goals of this work is to motivate the problem and solution from the ground up for a general non-medical or non-technical audience, the first portion of the thesis is dedicated to providing adequate background on the respiratory system, spirometry, sound and airflow physics, as well as machine learning. Consequently, these chapters do not cover the main technical contributions and can be skipped based on the reader's discretion. Following the background sections, the related work is covered along with the experiments that were conducted to guide the research. Next, a comprehensive coverage of the creation

and organization of the dataset is presented, followed by an outline of the Spiro AI system as well as the specifications for the proposed subsystems. Finally, the results are described followed by a conclusion which alludes to future work and the main insights.

2 Lung Background

In order to follow and appreciate the technical contributions outlined later in the thesis, a thorough exploration of the human respiratory system is required. In the following chapter, the evolution, anatomy, and physiology of the respiratory system will be covered in order to provide a foundation for understanding respiratory diseases and spirometry. The aim of this chapter is to provide a complete picture of how lungs function for a general reader who may not have a significant background in physics or biology.

2.1 Oxygen Miracle

> *Whether it be the sweeping eagle in his flight, or the open apple-blossom, the toiling work-horse, the blithe swan, the branching oak, the winding stream at its base, the drifting clouds, overall the coursing sun, form ever follows function, and this is the law. Where function does not change, form does not change.*
>
> **— Louis Sullivan**
> (19th-century skyscraper pioneer)

Leonardo da Vinci once proposed that air was made up of two gases; one for breathing and one for fueling fire. While his intuition was on the right track it is now known that these two gases are in fact one: oxygen, or O_2. Oxygen is the second most common gas found in the earth's atmosphere and third most common in the Milky Way; it plays a vital role in life on earth whether it is photosynthesis, respiration, or in the case of intelligent life, combustion. Oxygen gets its name from the French word oxygène meaning acidifying constituent. This name is fitting as oxygen is the second most electronegative atom in the periodic table after fluorine, and hence has strong tendency to rip electrons from other atoms. As a consequence, oxygen has a relatively short lifetime in the atmosphere [13]. Fortunately, due to photosynthesis, the supply in the atmosphere is continually replenished. Although oxygen appears plentiful in the atmosphere today, there was a time, before the evolution of photosynthesis where oxygen and subsequently life was scarce.

Prior to around 3.5 billion years ago, Earth's atmosphere and oceans were anoxic (lacking oxygen). This is supported by the discovery of sulfur isotopes in sediments from this time period which could only exist in the absence of oxygen [67]. Eventually, photosynthetic cyanobacteria emerged and created oxygen as a waste product deep in the ocean. This process is outlined by the photosynthesis equation in Figure 2.1 [34]. By about 2 billion years ago, this precious oxygen began to trickle into the atmosphere leaving a trail of rust along the ocean floor as a parting gift. This period of time from about 2.5 to 2 billion years ago is sometimes referred to as the Great Oxidation Event. Prior to this, respiration was anoxic, similar to anaerobic respiration, or fermentation. Unfortunately, at this time, oxygen was more of a toxin to life than a nutrient given its aggressive properties. It took a few ice ages and hundreds of million years for life to catch up and evolve an oxygen-tolerant metabolism.

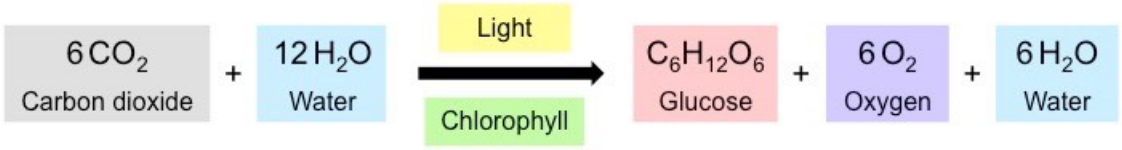


Fig. 2.1: The photosynthesis equation responsible for the majority of oxygen in the atmosphere

Oxygen, being as reactive as it is, can provide a large dose of energy to an organism, assuming the organism is prepared to deal with it. Subsequently, more energy leads to the evolution of larger and more advanced life. The spread of O_2 in the atmosphere, in turn, gave rise to O_3, which triggered the formation of the earth's protective ozone layer and allowed life to emerge from the ocean and colonize the land. As illustrated in Figure A of 2.2, the concentration of O_2 in the atmosphere peaked at over 30% around 350 million years ago and aside from minor oscillations, has stabilized at around 21%. Figure B of 2.2 shows the more recent trends of atmospheric oxygen as well as its high correlation to the climate change on Earth.

Even though oxygen is plentiful in our atmosphere, to many scientists, especially astrobiologists, the presence of atmospheric O_2 is rare enough to be considered a miracle. In fact, there are no known abiotic mechanisms that can produce an O_2 enriched atmosphere. In earth's case, a whopping 99.9999% of the oxygen in the atmosphere was produced by life [50]. As a result, if a planet with atmospheric O_2 is discovered, it is logical to conclude that life was the cause. Oxygen is perhaps the biggest contributor to Earth's uniqueness and the universal fuel that propels our rockets, cooks our food and powers the cells in our bodies. The focus of this work is on that of the respiratory system which serves as the carpool lane for delivering oxygen to our cells.

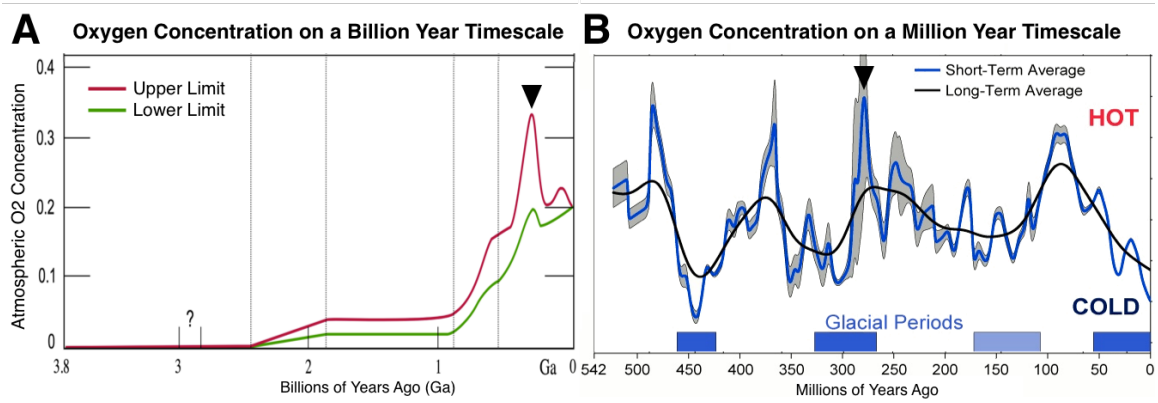


Fig. 2.2: Oxygenation of the atmosphere where in (A) is on a billion year timescale, and (B) is on a more recent million year scale. The Black triangle represents the same point on both plots, although the axis may be scaled differently, they both represent O_2 concentration

2.2 Evolution of Lungs

The purpose of a respiratory system is simple: *oxygen in, carbon dioxide out*. It elegantly compliments the photosynthesis process outlined in 2.1 which has the opposite effect of taking in carbon dioxide and outputting oxygen. Plants equipped with photosynthesis and animals with their respiratory systems are therefore entangled in a fruitful symbiotic dependence. But how could any living thing handle the acidic destructive properties of O_2 which cripples even the durable properties of iron?

2.2.1 Diffusion

It is difficult to know for sure how the aerobic proto-bacteria came into existence. What can be said is that they were around sixteen times more efficient than their anaerobic ancestors and as a result multiplied and dominated as bacteria do best [27]. This aerobic respiration is powered by a simple process known as diffusion, defined in Table 2.1. In this case, O_2 found in the water diffused into the bacterial cells and traded places with the CO_2 which was diffused out. The cell's phospholipid bilayer exceptionally facilitates this process by selectively allowing O_2 to enter and CO_2 to exit. What follows is somewhat typical for any dominant lifeforms, it consumes smaller, weaker competition and in turn grows stronger. Single-celled aerobic bacteria became mitochondria which developed intracellular compartments with specific functions. Finally, eukaryotic cells emerged which is what all multicellular animals are comprised of. At a cellular level, all aerobic life utilizes diffusion, but in order for organisms to grow into larger animals, the respiratory process needed continuous improvement.

Tab. 2.1: Relevant terminology for the evolution of lungs taken from the Oxford Dictionary

Term	Definition
diffusion	the process by which particles randomly and uniformly scatter from a high concentration area to a lower concentration area, requiring little to no energy.
gill	the paired respiratory organ of fish and some amphibians, by which oxygen is extracted from water flowing over its surface.
lung	the paired respiratory organ of most vertebrates and some fish, by which oxygen is extracted from external air pumped in by a process known as breathing.

2.2.2 Gills

While several fascinating flavors of respiratory systems have been studied, the gist of it can be conveyed through the evolution of aquatic to terrestrial animals. The available evidence suggests gills, defined in Table 2.1, were present in the very earliest fish and were responsible for the diffusion of oxygen. Since the amount of oxygen needed is correlated with body mass and diffusion is correlated with surface area, it is no surprise fish evolved into a form maximizing their surface area while minimizing volume. In general, the agility and efficiency of a fish are correlated with the size of its gills. Gills were great for a while, but about 350 million years ago, due to Earth's natural climate change cycle and the explosion of atmospheric O_2 (see Section 2.2), the oceans became shallower and fish needed a respiratory upgrade to survive. At some point, a fish known as the lobe-finned fish developed gas-filled organs that serve the function of respiration in addition to the already present gills. This is the first known species known to have lungs.

2.2.3 Lungs

Contrary to popular belief, lungs, defined in Table 2.1, did not evolve from the air bladders present in modern fish. It was actually the other way around [51]. The infamous lobe-finned fish was perhaps overly equipped with a double respiratory system, and today it is known that this is no longer a commonly occurring characteristic. So what happened to this redundant combo? Other than a few types of rare fish such as Coelacanths (dee Figure 2.3) and Lungfish which even today possess the lung gill combo, the gene pool somewhat forked.

Fish that continued to find refuge in the ocean did not need the complete lung package. Subsequently, their lungs evolved into swim bladders which simply held gas and helped the

Fig. 2.3: A rare coelacanth fish which processes lungs, gills and fins that evolved into land ready legs. Until it's recent re-discover, it was thought to be over 350 million years old and extinct for 65 million years.

fish control its buoyancy, but did not contribute to respiratory functionality. This type of fish is the common ancestor for the majority of fish present today. In contrast, the more unique fish with the lung gill combination became known as a tetrapods, meaning four feet, and took advantage of their strong fins and newly processed lungs, that enabled them to make their way out of the water onto land. Some tetrapods like frogs kept their amphibious abilities while others, such as lizards traded their gills in for thicker skin and stronger, land optimized legs. Tetrapods are the common ancestor of all terrestrial animals including reptiles, birds, mammals, and humans. We owe our entire existence to the evolution of lungs and the courageous tetrapods that carried them out of the water to land. When organisms more fully devoted to air breathing are observed, it is apparent the anatomy of their respiratory system is considerably more sophisticated, as illustrated in Figure 2.4.

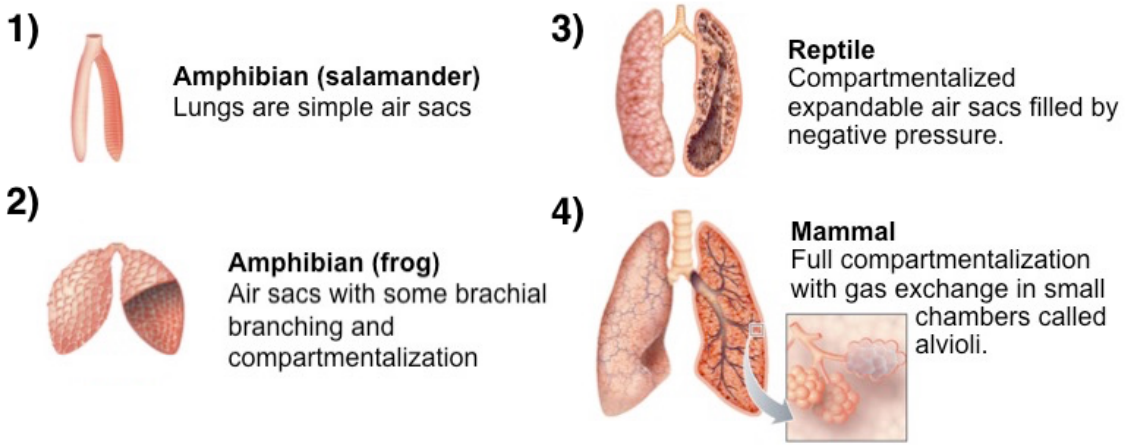


Fig. 2.4: Key stages in the evolution of lungs in evolutionary order from 1 to 4

As mammals grew, they required more oxygen which in turn required a larger, more robust and efficient respiratory system. Due to this necessity, mammals developed a strong diaphragm and rib cage, as well as a rigid trachea to support and aid their growing lungs. By the time humans emerged in the evolutionary chain, the mammal respiratory system experienced many beneficial upgrades. Along the way, several other enhancements emerged for different species solving different functional requirements. For example, in order for birds to breath at altitudes above the Himalayas, the avian lung evolved to perform continuous ventilation powered by the same muscles that flap their wings [51]. While several other interesting examples exist, the remainder of this work will focus on the human respiratory system. The next section will cover the physiology of human lungs and introduce key anatomy.

2.3 Physiology of the Respiratory System

The following section will describe the components and functionality of the human respiratory system. In section 2.3.1, the basic anatomy will be covered and more specific functional and physical details will be in the sections following.

2.3.1 Anatomy

The respiratory system is a complex biological system comprised of several organs facilitating the inhalation of oxygen and exhalation of carbon dioxide among other things. For the most part, respiration is handled by the lungs, but several other components are critical for the complete functionality, namely, the nose, mouth, pharynx, larynx, trachea, bronchi and bronchioles, and respiration muscles. In this section, a brief overview of these components will be presented from the head moving down to the torso. See Figure 2.5 for a visual reference. The following section is based on figures and definitions are from the National Heart Lung and Blood Institutes(NHLBI) online educational material [35, 72].

Nose and Nasal

The nose and nasal cavity constitute the main external opening of the respiratory system. They represent the entryway to the respiratory tract. Although the nose is typically credited as being the main external breathing apparatus, its main role is to protect and support the downstream respiratory processes. The windy passage is lined with mucus membranes and small hairs that filter the air before it enters the respiratory tract, trapping harmful particles such as dust, mold, and pollen.

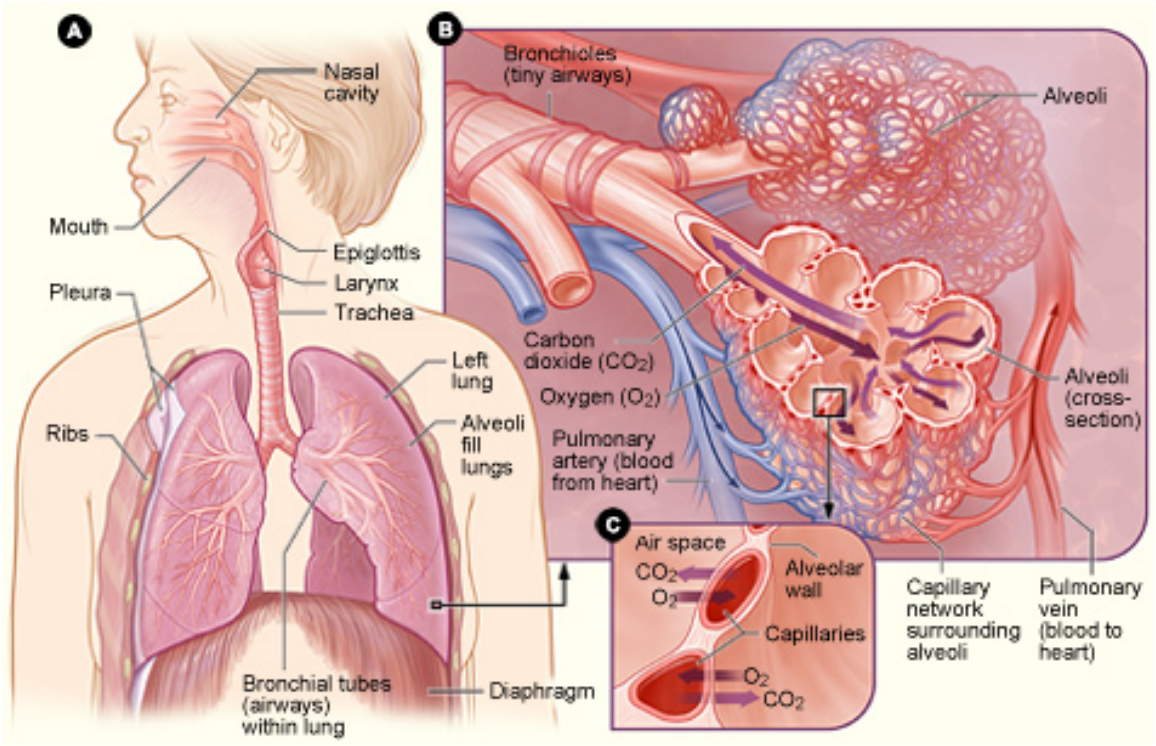


Fig. 2.5: (A) shows the respiratory system anatomy. (B) is an enlarged view of the airways, alveoli, and capillaries. (C) is a closeup view of gas exchange between the capillaries and alveoli [35].

Oral Cavity

The oral cavity, or mouth, is the only other external component of the respiratory system. It provides similar functionality to the nasal cavity and acts as a supplement or alternative to the air inhaled through the nose. Unlike the nasal passage, the mouth does not possess the mucus or small ciliary hairs capable of filtering out particles. Instead, it is a direct path for large bursts of input and output airflow due to its larger diameter and direct path.

Pharynx

The pharynx, or throat, is the next component of the respiratory tract. It resembles a funnel made out of muscles acting as an intermediary coupling between the nasal cavity and the larynx and esophagus. It houses the epiglottis which is a flap that performs the vital task of switching access between the esophagus and trachea. This ensures air is routed through the trachea, and the ingested food is diverted to the esophagus.

Larynx

The larynx represents a small section of the respiratory tract that connects the bottom of the pharynx to the trachea. It is commonly referred to as the voice box and contains thyroid cartilage, or Adam's apple, cricoid cartilage and the vocal folds. Both cartilages offer protection and support to other more sensitive components such as the vocal cords. The

vocal cords are comprised of mucus membranes that tense up and vibrate, creating sound and speech.

Trachea

The trachea, or windpipe, connects the larynx to the bronchi. It is a more rigid section of the tract and is shaped like a corrugated tube approximately 5 inches in length. It has several hyaline cartilage rings that keep the trachea open and prevent it from collapsing in on itself due to the negative pressure encountered during inhalation. The hyaline cartilage is actually C-shaped such that the open end faces the esophagus, permitting the esophagus to expand into the trachea when larger pieces of food are swallowed. The trachea is lined with mucus-producing epithelium and velcro-like cilia, which traps particles in the incoming air and prevents them from gaining entrance to the lungs.

Bronchi

The lower end of the trachea splits the respiratory tract into two branches called the primary bronchi. These pass through the top of the lungs and then branch into smaller bronchi. These secondary bronchi continue carrying the air to the lobes of the lungs, then further split into tertiary bronchi which further continue into what are called terminal bronchioles. This inverted tree structure effectively tries to maximize its coverage within the lung similar to how leaves of a tree attempt to maximize the surface area facing the sun. A single lung has millions of terminal bronchioles less than a millimeter in length which directly deliver oxygenated air into the alveoli. The larger bronchi contain C-shaped cartilage rings to keep the airways open similar to those found in the trachea. In contrast, the tiny bronchioles rely on flexible muscles and elastin to keep form. Also, like the trachea, the bronchi and bronchioles are lined with mucus and cilia to trap any foreign particles.

Up to this point, the components described perform a similar respiratory function of moving air to and from the lungs and therefore can be considered broadly as airways. These airways can be classified as extrathoracic, as in outside the lungs or intrathoracic, or within the lungs.

Lungs

The lungs are two organs located inside the thorax on the left and right sides, weighing together about 3 lbs and occupying roughly the same surface area as a tennis court. They are surrounded by a membrane that provides them with enough space to expand when inflated with air. Due to the location of the heart, the lungs are not symmetrical. The left lung is smaller and has only 2 lobes while the right lung has 3. The inside of lungs resembles a sponge made of about 500 million small sacs called alveoli. These alveoli are found at the ends of terminal bronchioles and are surrounded by capillaries through which blood is routed. The lumen of these capillaries are so small that individual red blood cells are forced

to line up in single file as they pass the alveoli. The epithelium layer covering the alveoli performs the gas exchange with the blood flowing through the capillaries.

Respiratory Muscles

The muscle structure known as the respiratory muscles surround the lungs and permit the inhalation and exhalation of air. The diaphragm is the main muscle in this system and consists of a thin sheet of muscle that forms the floor of the thorax. It pulls air into the lungs by contracting several inches with each breath similar to the plunger in a syringe being pulled back. In addition to the diaphragm, multiple intercostal chest muscles are located between the ribs and also aid the lungs in compression and expansion.

Conclusion

The ancient Greeks can be thanked for creating these interesting and unintuitive terms. Like the Greeks, modern physicians, lawyers, and physicists tend to use large words, for reasons up for debate. If up to a modern engineer, the names of the components would be much less inspiring; perhaps input/output tubes (oral, nasal), coupler (pharynx, larynx), rigid pipe (trachea) and manifold (bronchi), tank (lungs), energy converters (alveoli). Put this way, the respiratory system does not seem so foreign and is far more analogous to an automobile in that the engine intakes and exhausts air and other gases to perform its energy conversion via combustion. Nature and human engineering have many commonalities which stem from the logical nature of assembling something from the ground up to perform a specific function. In the next section, the operation and design of the respiratory system will be discussed from a functional point of view.

2.3.2 Functional requirements

Functional morphology involves the study of relationships between the structure of an organism and the function of the various parts. The quote from the beginning of this chapter, "form ever follows function", is a guiding principle of functional morphology. In biology, the idea of relating form and function originated with the French naturalist Georges Cuvier (1769-1832) and was later elaborated upon by Charles Darwin. Because evolution occurred on a timescale well beyond recorded history, it is near impossible to know the complex mapping between a needed function and resulting biological component. Fortunately, due to today's rich biodiversity, similar components like the lungs, for example, can be studied between species. The similarities in form may allude to a universal set of functions, while differences may uncover species-specific function. In biology, these functions usually boil down to staying alive in various unpredictable environments. For example, all species with lungs have a way of creating positive and negative pressure to breathe, but the way this function manifests is quite different depending on the type of animal. Scaled reptiles use the same muscles to both move and breath, which means they, unfortunately, can only

do one or the other. Mammals, which enjoy much more mobility, come equipped with a diaphragm muscle for breathing which can be used independently of the limb muscles. The low mobility in reptiles may make breathing less of a ubiquitous activity, but the sacrifice rewards them with sharp claws and heavy armored skin. Engineering is also driven by a functional, axiomatic design methodology. Similar observations arise when comparing the material used in a tank versus a sports car. Rather than simply stating the facts making up *what* the components do and *where* they are located as in the previous 2.3.1 section, this section will attempt to answer the more difficult questions of *why* the human respiratory is the way it is and *how* it works. Much of the content in this section is summarized from the 1988 article: Form and function of lungs[51].

Diffusive Medium

The cardinal function of the lung is gas exchange via the passive process of diffusion. The metabolic waste product carbon dioxide is delivered by the circulation system to the alveoli, where it is exchanged for fresh oxygen delivered via airways during inhalation. To best support this, the diffusive contact medium between air and blood must have the properties of maximal surface area (high flux) and minimal material (low resistance). Alveolar type I epithelial cells, which make up the medium where diffusion occurs, perfectly fit this requirement. Type I epithelial cells are utilized for diffusion because they are extremely thin, flexible and modular, allowing them to occupy large, complex surface areas while also enabling many diffusion pathways to maximize throughput. These cells line 80-90% of the alveolar surface. Replacing the surface level skin diffusion found in primitive life with internal breath powered diffusion mechanisms such as those found in alveoli is similar to upgrading a sidewalk to a highway in the sense that both quantity (faster speed) and efficiency (several lanes) are optimized.

Minimal Surface Tension

As the lung became increasingly efficient in terrestrial vertebrates, alveoli became progressively smaller and more abundant, but at the cost of being more fragile. In order to prevent these tiny bubble-like alveoli from popping, the surface tension of the air-water interface the alveoli are immersed in must be minimized. The solution is another cell type, the type II epithelial cell which secretes a foam-like substance known as surfactant. While type II epithelial cells occupy only a small fraction of the alveolar surface, the surfactant they produce is plentiful and crucial in reducing the surface tension. Furthermore, this foamy medium provides an additional layer of protection to the sensitive alveoli.

Elastic Bag

From a functional point of view, in order to properly ventilate, the lung must behave like an elastic bag capable of moving freely to allow expansion and contraction of all its parts. This elasticity must be able to expand to fill available space and then contract without completely collapsing on itself (unlike a balloon). The mammalian lung meets this functional

requirement by employing a mixture of elastic fibers and collagen that have different mechanical properties: elastic fibers are extensible up to about 130% their relaxed length and inherently possess a useful recoil force. The collagen fibers, however, are inextensible and have very high tensile strength to give the lung an acceptable degree of stiffness in order to prevent too much contraction. This mixture of fibers yield an ideal balance between elastic recoil and tensile strength, providing a framework well matched with the functional demand to allow repeated and rapid contractions and expansions. One weakness of this design is lungs lack any sort of protective outer tissue layer making them vulnerable to blunt force and sharp objects. Fortunately, the rib cage which surrounds the lungs and other nearby vital organs provide sufficient protection.

Automatic Maintenance

The lung surface, which is continuously exposed to our environment and made of a mosaic of as many as 40 different cell types including the ones described above, must be continuously cleaned and maintained in order retain high-efficiency diffusion. The combination of mucus and ciliated cells perform these functions and are found throughout the airways where large volumes of air flow into the lungs; regions most susceptible to foreign objects such as dust. In advanced mammals, the process by which foreign material is removed is referred to as the "mucociliary escalator" and it is quite elegant despite its unappealing name. The ciliated cells lining bronchial tubes and trachea have a claw-like structure that catches any foreign objects that would otherwise progress deeper into the lung. The mucus forms a layer that flows up the trachea due to an upwards beating effect caused by the cilia. As a result, this mucus and any debris caught by the ciliated cells are forced up and out of the respiratory system, hence the name "mucociliary escalator". Normally the bronchial mucus is flushed into the pharynx and swallowed unnoticed, however, when mucociliary escalator becomes inactivated by perhaps nicotine or excessive dust, it receives assistance in the form of coughing. This process also warms and moistens incoming air to better prepare it for efficient diffusion.

Mobility Enhancement

As mentioned earlier, unlike reptiles, mammals can breathe while doing other heavily aerobic tasks such as running and hunting. The ability to perform both of these in parallel gives mammals a significant advantage and can be attributed to their dominance today. Humans and other mammals take this advantage to an extreme and develop ways of allowing mobility to enhance breathing. For example, as seen in the cantor of a horse, inhalation is coincident with the lifting of the front limbs, which naturally pulls in air. When the limbs return to the ground, the rib cage undergoes compressive force which forces air to abruptly exhale in time for the next step. Human sprinters are familiar with these breathing harmonies as they utilize them in order to achieve top performance. This example highlights how species can improve survival by forming new functional requirements that are eventually baked into the genetic code by natural selection. In this case, two independent and contradicting

operations, running and breathing have fused together to create a single, synergistic system that serves multiple functions.

Conclusion

This section provided the key functional requirements met by the ingenious design of the human respiratory system. The final topic for this chapter covers the mechanics powering the respiratory system, which completes the physiology of the respiratory system and serves as the foundation for understanding Chapter 3 which explores common respiratory diseases that are often a result of a disturbance or limitation in the system.

2.3.3 Mechanics of Breathing

Breathing, or pulmonary ventilation, is the process by which air flows into the lungs during inspiration (inhalation) and out of the lungs during expiration (exhalation). Like all gases, air flows from a region of higher pressure to a region of lower pressure and it is the pressure difference between the atmosphere and the gases within the lung that permits breathing. Muscular breathing movements and elastic tissue recoil are the main sources that contribute to the pressure changes within the lung.

Inspiration

Inspiration is considered the active phase of ventilation because it is the result of muscle contraction. During inspiration, the diaphragm and other muscles contract and the chest cavity increases in volume in both the lateral and the anteroposterior (front to back) directions, similar to expanding a bellows. This causes negative pressure in the lungs and forces the intake of air. Bernoulli's Principle states when the speed of gas increases, the pressure decreases, thus conserving energy. In the case of inspiration, the incoming air causes a pressure drop in the extrathoracic, upper airways, causing constriction. Alternatively, the intrathoracic airways within the lung expand as air fills the lungs.

Gas Exchange

Once the air reaches and enters the alveolar sacs, oxygen from inspired air diffuses across the very thin epithelial wall of the alveoli to the adjacent capillaries. A red blood cell protein called hemoglobin helps transport oxygen from the air sacs to the blood. Simultaneously, carbon dioxide moves from the capillaries into the air sacs to be expelled during exhalation. On a broader scope, the oxygen-poor blood being delivered to the alveolar structures comes from tissues throughout the body. This blood returns to the right side of the heart and is pumped via the pulmonary artery to the lungs where the critical gas exchange takes place and oxygen eagerly trades places with the metabolic waste product, carbon dioxide. The oxygen-rich blood in the alveolar capillaries returns through the pulmonary vein to the left side of the heart which then pumps the oxygen-rich blood to the rest of the body.

Expiration

Unlike inspiration which requires energy and muscular effort, expiration is very efficient and being passive, adds no extra physiologic cost. During expiration, the diaphragm simply relaxes which triggers the elastic lung tissue to recoil and subsequently the chest cavity volume decreases. This increases the pressure within the lungs and pushes air back out to the atmosphere. The airways undergo an opposite effect to inspiration, namely the intrathoracic airways shrink while the extrathoracic counterparts expand.

When a person is physically active, abdominal muscles contract and push the diaphragm against the lungs even more than usual. This rapidly pushes air out of the lungs but is no longer passive as it requires extra energy.

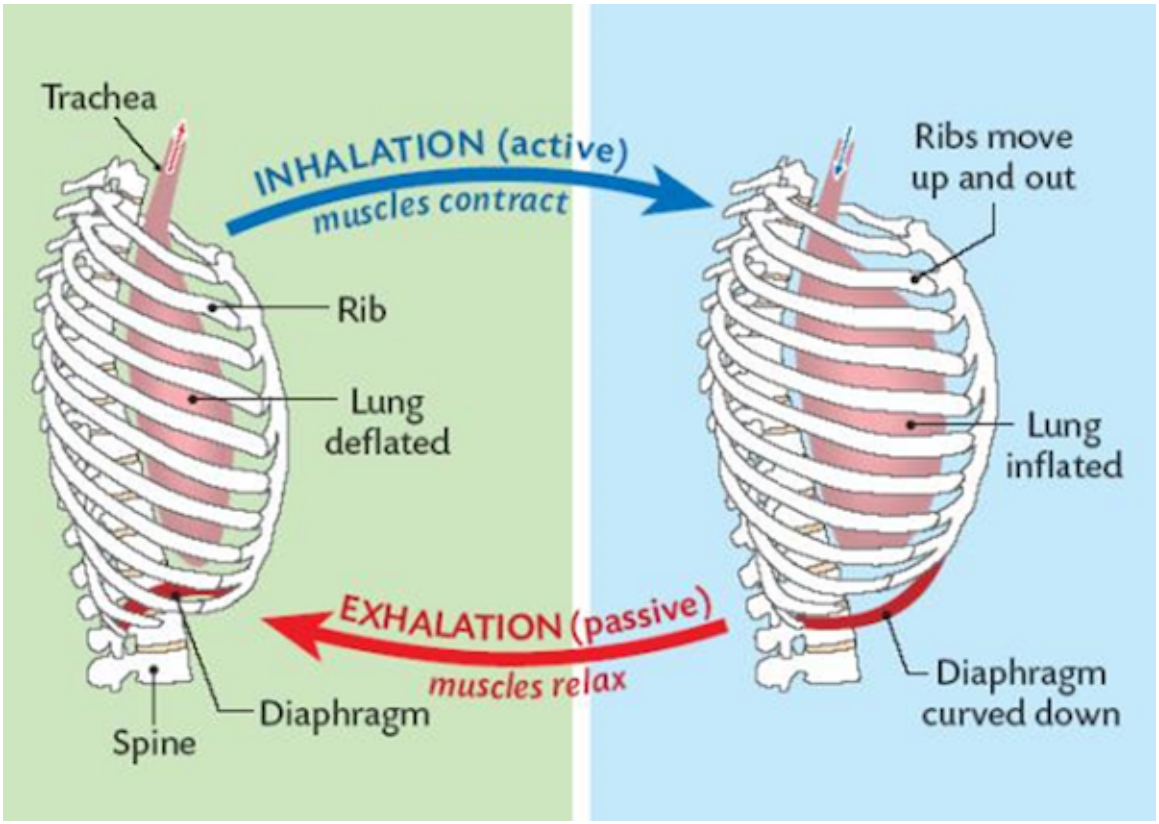


Fig. 2.6: Mechanics of ventilation involve a cycle of inhalation and exhalation.

Energy Conservation

From a physics perspective, the potential energy created by the contraction of the diaphragm is temporarily stored in the elastic tissues of the lung and chest muscles. Like a loaded spring, this energy is released when lung and chest muscles recoil, causing exhalation as illustrated in Figure 2.6.

Control Mechanism

Most complex systems follow the standard control flow: a control signal triggers a particular action, sensors observe a change in state due to the action and report feedback which is then used to define the next control signal to yield the next desired state. In the advanced life of mammals, the brain acts as the control signal generator to trigger muscle action. It relies on sensors such as nerves, eyes, and ears to perceive the environment and choose actions to optimally reach the desired state.

Respiratory muscle control works like most other voluntary and involuntary muscle control, the signal originates in the brain and propagates to the destination muscles via the spine. Respiratory control happens unconsciously to ensure breathing muscles contract and relax regularly and automatically. To a limited degree, this control can be overridden, for example, ones breathing rate can be altered consciously by breathing faster or by holding one's breath. Emotion, stress and physical activity can also affect breathing control.

There are a number of sensors in the brain, blood vessels, muscles, and lungs that provide crucial feedback to the brain's control strategy. Sensors in the brain and in major blood vessels detect carbon dioxide or oxygen levels in the blood and change your breathing rate as needed. Other sensors in the airways can detect irritants and trigger actions such as sneezing and coughing. The alveoli are also equipped with sensing capabilities that can detect fluid buildup in the lung tissues which are thought to trigger rapid, shallow breathing. Finally, sensors in joints and muscles detect movement of your arms or legs and may play a role in increasing your breathing rate during physical activity.

2.4 Conclusion

The goal of this chapter was to provide the reader with a sufficient understanding of *why* humans have lungs, *how* they work and *what* the system is comprised of. For readers new to respiratory science, hopefully, this provided sufficient background to understand the upcoming chapters. For those who consider themselves advanced in the topic, perhaps the broad and multidisciplinary overview provided a newfound appreciation and insight into the evolution and function of the respiratory system.

3 Respiratory Disease

This chapter will explore the primary types of respiratory diseases, the common causes and risk factors, as well as the treatment used to alleviate the symptoms. Following this, the relevant epidemiology will be summarized in order to highlight the critical importance and necessity for pervasive screening and diagnostic tools for respiratory disease.

3.1 Types of Respiratory Diseases

Lung diseases can be categorized into four general types: *restrictive*, *obstructive*, *ventilation* and *perfusion* related disorders. In simple terms, *restrictive* means something restricts air from filling the lungs, *obstructive* means something is obstructing airflow out of the lungs, *ventilation* means something is preventing the gas exchange process from adequately functioning, and *perfusion* means something is compromising the blood supply to or from the lungs. The focus of this work is on restrictive and obstructive diseases as these forms impact the most people and are typically diagnosed with spirometry based lung function tests, which are outlined in the upcoming Spirometry chapter. Restrictive and obstructive diseases are formally defined in Table 3.1; both share the same main symptom of shortness of breath upon exertion, but obstructive lung disease is far more commonly encountered. While there is no single cause for lung disease, the most common contributors include cigarette smoking, air pollution, infections, or genetics.

Tab. 3.1: Definitions of the two main lung disease categories according to WebMD

Term	Definition
restrictive	Restrictive lung disease can make it difficult to fully fill lungs with air due to some form of lung restriction. Such restrictions are often caused by conditions causing stiffness in the lungs themselves or in other cases, stiffness of the chest wall, weak muscles, or damaged nerves.
obstructive	Obstructive lung disease causes shortness of breath due to difficulty exhaling all the air from the lungs. It can be a result of damage to the lungs or narrowing of the airways inside the lungs which can cause air to exhale much slower than normal.

Traditionally, the majority of the research and clinical attention has emphasized the obstructive group of diseases as they are by far the most prevalent; however, the work described in this thesis can be utilized as a diagnostic and trend reporting tool for both obstructive and restrictive diseases. Subsequently, both are considered. There are similarities between restrictive and obstructive disease. As mentioned, they both have the common symptom of shortness of breath, although for very different reasons. Coughing is also a common clinical manifestation observed in restrictive and obstructive lung diseases. Usually, the cough is dry or productive with white or colorless sputum. The frequent use of anti-inflammatory medicines and supplemental oxygen to manage restrictive and obstructive lung disease is another common feature shared by both conditions. Beyond this however, the causes and other treatments employed are very dependent on the specific disease and how it manifests itself in the respiratory system. The following sections will outline resitrictive and obstructive disorders, along with treatments options based on information provided by WebMD and the NHLBI[35, 53]

3.1.1 Restrictive Diseases

Restrictive lung diseases are characterized by reduced lung volumes; the ability of the lungs to fully expand is diminished. They can be grouped into two anatomical categories: *intrinsic* describes diseases occurring within lung, while *extrinsic* diseases occur outside the lungs. Within these two categories there are over 200 known causes, making treatment difficult. For example, fibrosis, causes the lung tissue to harden, making it very difficult for the lungs to expand and intake air. Obesity or scoliosis, on the other hand, cause mechanical restriction by squeezing the lungs which also impedes the lungs ability to expand. In most cases a patient with a restrictive disease has to exert extra energy to intake air, but due to the restrictive nature of the lungs, there is no place for the air to go. So more work is exerted with less of a reward.

Intrinsic

Intrinsic lung diseases cause inflammation or scarring of the lung tissue (interstitial lung disease) or result in filling of the air spaces with debris (pneumonitis). With the wide variety of different causes of restrictive disease, it is often difficult to pin-point a specific cause. When there is no known cause, the umbrella disease, idiopathic pulmonary fibrosis (IPF) is diagnosed by default. Roughly 60% of patients fit into this category. When the cause is identified, it tends to be one of the following: connective-tissue diseases, drug-induced lung disease, environmental exposures (inorganic and organic dusts), or inflammatory lung diseases such as sarcoidosis.

Figure 3.1 illustrates the effect of asbestos induced pulmonary fibrosis on the terminal bronchi and alveoli which facilitate diffusion. In the normal lung (Figure A of 3.1), the

space between the alveoli and blood supply is very small, on the order of 0.2 μm, which is about 7/1,000,000ths of an inch. This enables oxygen to diffuse efficiently and very quickly (roughly 0.75 seconds). In contrast, the lung tissue affected by fibrosis (Figure B of3.1), has a thickened membrane. Although diffusion can still occur, the increased thickness and density of the membrane between the air and blood supply greatly reduces the speed and efficiency of diffusion, which in turn reduces the effectiveness of each breath. To further complicate matters, over time patients with IPF replace their normal elastic lung tissue with stiff fibrotic scar tissue which is much less elastic. The end result of IPF is damaging on two fronts: 1) less air can come in and out, and 2) less of the oxygen in the air that does make it into the lungs is diffused into the blood.

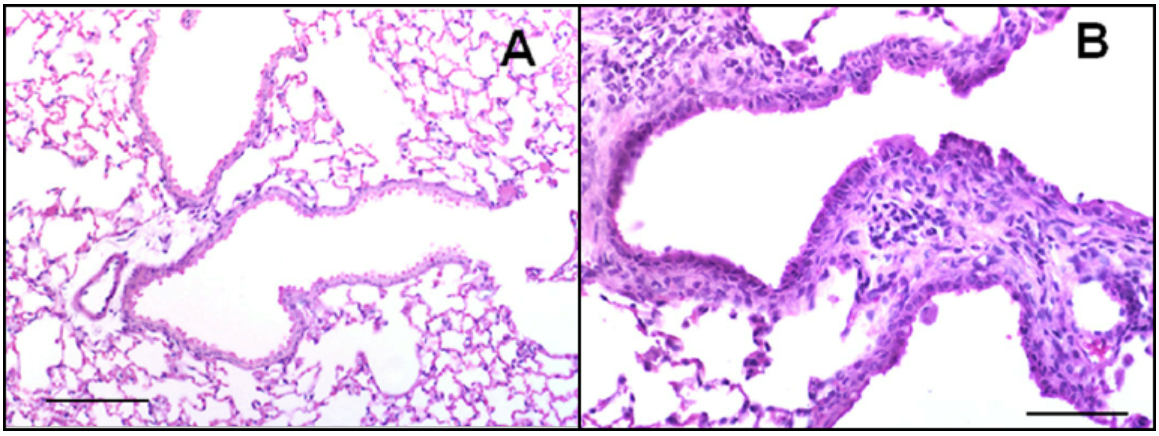


Fig. 3.1: Asbestos induced pulmonary fibrosis. (A) Control lung shows normal terminal bronchi and alveoli. (B) Intratracheal instillation of crocidolite asbestos induces fibrosis (14 days after exposure) [6].

The top five most common intrinsic restrictive disorders include:

- Idiopathic pulmonary fibrosis
- Interstitial lung disease
- Pulmonary Fibrosis
- Sarcoidosis
- Pneumoconiosis

Extrinsic

Extrinsic or extra-pulmonary respiratory diseases effect the exterior components responsible for ventilation of air, such as the chest wall, exterior lung tissue, and respiratory muscles. These diseases can be neuromuscular (polio), nonmuscular (scoliosis), or due to foreign material like asbestos trapped between the chest wall and lung exterior. Imagine being bear hugged by Dwayne "The Rock" Johnson and also trying to take a deep breath.

The top five most common extrinsic restrictive disorders include:

- Obesity
- Pleural Effusion
- Myasthenia gravis
- Scoliosis
- Neuromuscular disease, such as muscular dystrophy or Lou Gehrig's Disease (ALS)

3.1.2 Obstructive Diseases

Imagine taking a deep breath and then trying to exhale through a drinking straw. This is similar to what a patients with obstructive lung disease deals with on a regular basis. Obstructive lung disease makes it difficult to exhale old CO_2 rich air from the lungs because of the narrowing of the airways, or forms of lung damage. Exhaled air is expelled more slowly than normal and at the end of a full exhalation, an abnormally high amount of air may still remain trapped in the lungs. Obstructive lung disease makes breathing especially harder during increased activity or exertion. As the rate of breathing is increased and the lungs work harder, the amount of fresh air circulated through the lungs is decreased because obstructed exhalation cannot keep up. This results in hyperinflated lungs with too much stale CO_2 and not enough fresh O_2. Over time hyperinflation can result in a more permanent clinical feature known as "barrel chest", which describes a chest with a large front-to-back diameter.

Unlike restrictive diseases which have hundreds of potential causes, the most frequently encountered conditions associated with obstructive diseases are far more limited and include:

- COPD
- Asthma
- Bronchiectasis
- Cystic fibrosis

These conditions, which may exist simultaneously, are outlined in the following subsections.

COPD

Chronic obstructive pulmonary disease (COPD) is an irreversible, progressive chronic inflammatory lung disease encompassing several conditions, most commonly emphysema and chronic bronchitis. It is by far the most deadly respiratory disease as it is the 3rd (and rising) cause of overall death with an estimated economic cost of $2.1 trillion in 2010 [52]. Less air flows in and out of the airways because of one or more of the following: reduced elasticity

of airways and alveoli, destruction of alveolar walls, inflamed and thickened airways with excessive mucus production.

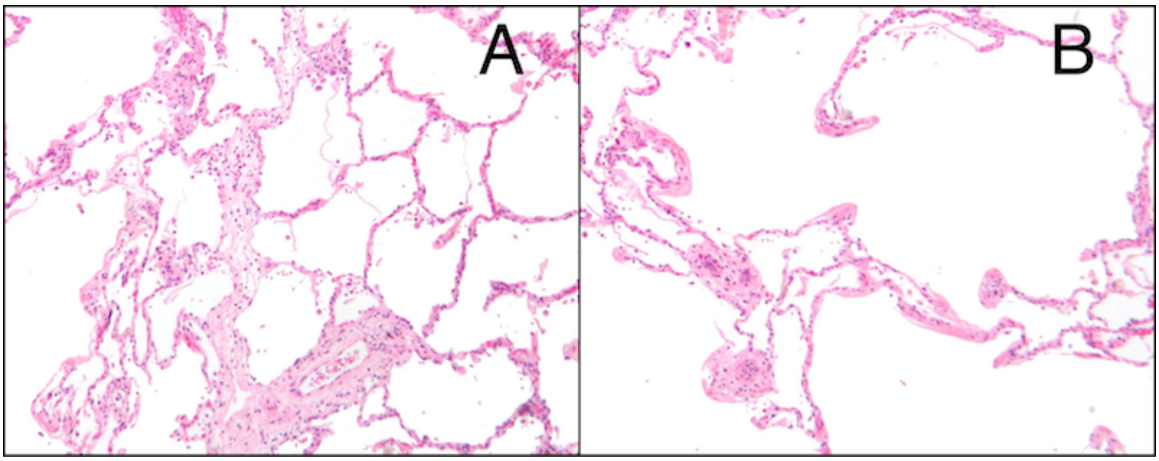


Fig. 3.2: (A) is a healthy lung with most of the alveoli still intact. (B) shows a lung with emphysema which characteristically has more open space once occupied by now collapsed alveoli.

Emphysema is a condition in which the alveoli walls are destroyed as a result of damaging exposure to cigarette smoke and other irritating gases and particulate matter. As a result, the air sacs lose their integrity and collapse, leading to fewer and larger air sacs instead of several tiny efficient ones. The micrograph in Figure 3.2 shows the preserved alveoli in the healthy lung (A) versus an emphysemic lung (B) which has large, ill-defined spaces secondary to collapsed alveoli.

Chronic bronchitis targets the larger airways and involves inflammation of the lining of the bronchial tubes. It is characterized by daily cough and excessive mucus production. In chronic bronchitis, the lining of the airways stays constantly irritated and inflamed, causing airway swelling. As a result thick mucus forms in the airways, further obstructing the airways and making it hard to breathe.

Most people who have COPD have both emphysema and chronic bronchitis; however, the severity of each condition varies from person to person, thus, the collective term COPD is more accurate. Figure 3.3 illustrates the effects of both emphysema and bronchitis.

COPD is caused by long-term exposure to irritating gases or particulate matter. Most people who have COPD smoke or used to smoke; however, up to 25% percent of people with COPD never smoked. Longterm exposure to other lung irritants such as air pollution, chemical fumes, or dusts can also contribute to COPD. A rare genetic condition called alpha-1 antitrypsin (AAT) deficiency can also cause the disease. Other respiratory diseases such as asthma can also progress into COPD if left untreated long enough.

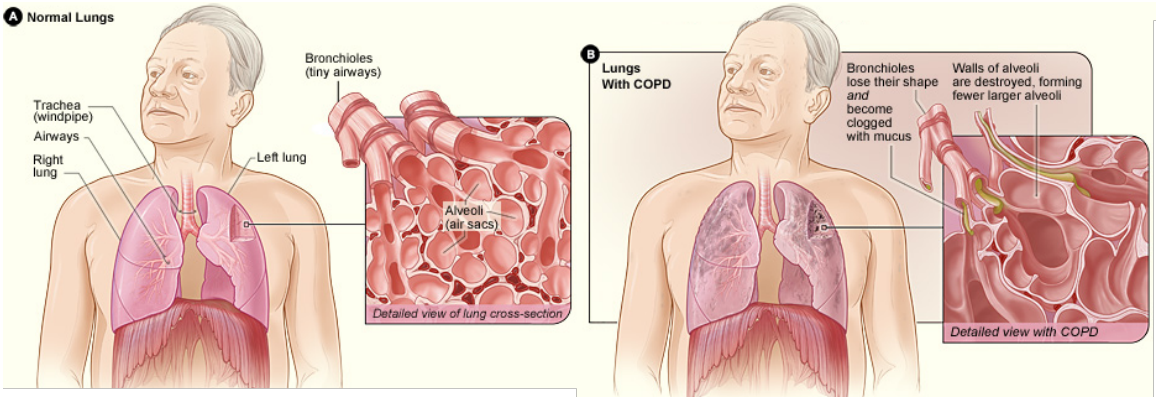


Fig. 3.3: (A) shows example of healthy lungs. The inset image shows a detailed cross-section of the bronchioles and alveoli. (B) shows lungs damaged by COPD, including damage to the bronchioles and alveolar walls [35].

Symptoms of COPD include shortness of breath, chronic cough, , wheezing, excess sputum production and chest tightness. People with COPD are at increased risk of developing heart disease, lung cancer and a variety of other serious conditions. Since COPD develops gradually, its progression can be slowed or prevented by minimizing exposure to risk factors, such as smoking. It is usually diagnosed in middle-aged or older adults. While there is no cure or way to reverse the damage, COPD is treatable. With proper management, most people with COPD can achieve good symptom control.

Asthma

Asthma is a chronic lung disease associated with inflamed and narrowed airways, which makes breathing difficult and triggers coughing, wheezing and shortness of breath. For many people, asthma is a minor nuisance. For others, it can be a major health problem that interferes with every day activities and may lead to life-threatening acute asthma attacks. The symptoms may flare-up or be more active in the morning or at night.

The inflammation caused by asthma makes the airways swollen and sensitive, causing them to react strongly to certain inhaled particulates. When the airways react, the muscles tighten, further narrowing the lumen of the airways and allowing less air to enter the lungs. This also causes the cells in the airways to generate more mucus than usual, further blocking the airways. This effect is illustrated in Figure 3.4(1).

Asthma affects people of all ages, but it most often starts during childhood. In the United States, more than 25 million people are known to have asthma. About 7 million of these people are children, making it the most common non-communicable disease among children [5].

The exact cause of asthma is not known. Researchers believe both genetic and environmental factors contribute to the development of asthma, usually early in life. These factors include: parents with a history of asthma, presence of allergies, or early childhood respiratory infections that manifest while the immune system is still developing. While it can not be cured, asthma symptoms can be adequately controlled. Treatment can typically reverse the inflammation and narrowing occurring due to asthma. Rescue inhalers are used to treat acute symptoms and maintenance inhalers are employed to prevent symptoms. Severe cases may require longer acting inhalers and oral steroids to counteract the inflammation and keep the airways open. Because of these treatment options, most people who have asthma are able to effectively manage the disease and end up living healthy, active lives. Not long ago, asthma was included in COPD, but since it is episodic and reversible, it has come out from under the COPD umbrella. That being said, asthma can advance to COPD if left untreated or if poorly managed.

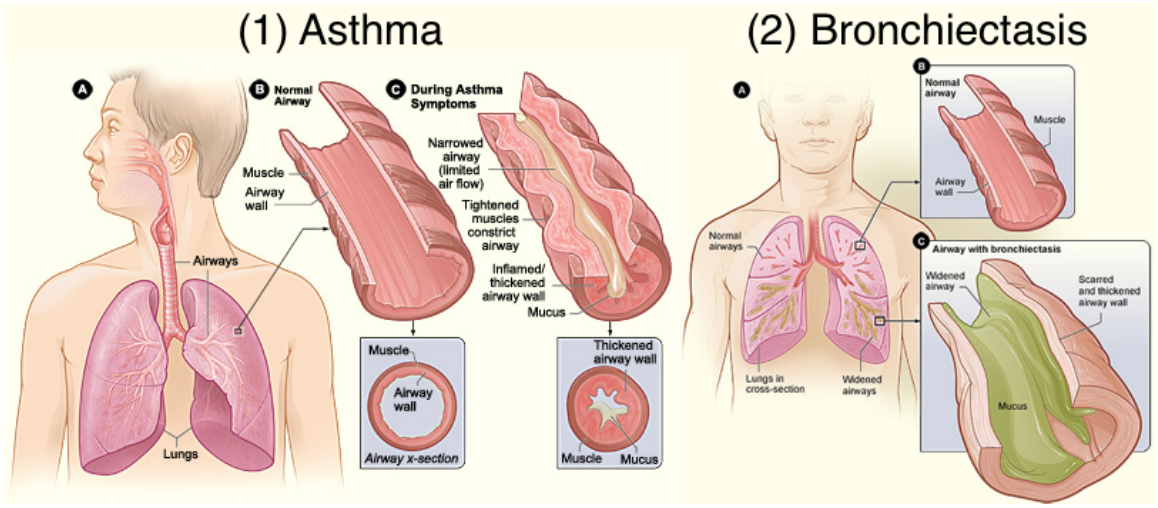


Fig. 3.4: (A) shows example of healthy lungs. The inset image shows a detailed cross-section of the bronchioles and alveoli. (B) shows lungs damaged by COPD, including damage to the bronchioles and alveolar walls [35].

Bronchiectasis

Bronchiectasis is a condition in which damage to the airways causes them to widen and become loose and scarred. It is usually the result of repeated infection or other conditions that injure the airway walls or prevents the airways from effectively clearing mucus. See Figure 3.4 (2) for an illustration. When mucus cannot be cleared, it builds up, creating an environment in which bacteria can thrive. This leads to repeated, serious lung infections causing irreversible airway damage. Eventually, bronchiectasis can lead to serious health problems, such as respiratory failure, atelectasis, and heart failure due to lack of inadequate oxygen intake.

Common childhood infections such as whooping cough and measles used to be responsible for many cases of bronchiectasis. However, due to modern vaccinations and antibiotics avail-

able in developed countries, these causes are now much less common. Instead, bronchiectasis usually is due to a medical condition or infection that injures the airway walls or interferes with the airways ability to clear mucus. Examples include infections such as severe pneumonia or tuberculosis, and conditions such as cystic fibrosis, immunodeficiency disorders and primary ciliary dyskinesia. Bronchiectasis doesn't always affect both lungs. When only one part of the lung is affected, the cause is typically attributed to a blockage rather than a medical condition. Congenital bronchiectasis, while less common, stems from a defect occurring during lung development of the fetus.

Cystic fibrosis

Cystic fibrosis (CF) is an inherited genetic disease of the secretory glands which produce mucus and sweat. People with CF must inherit two faulty genes, one from each parent; therefore, it is likely the parents do not have the disease themselves. CF affects the whole body, including the lungs, pancreas, liver, intestines and sinuses, but the focus here will be on the lungs.

CF leads to almost half of the cases of bronchiectasis in the United States because it causes mucus to be excessively thick and sticky, leading to airway blockage. Subsequently much of the bronchiectasis description above applies to CF. Since CF also affects the entire body, there are many other detrimental effects of the disease such as digestive and malnutrition issues, osteoporosis, infertility and imbalances in blood minerals to name a few.

3.1.3 Common Lung Disease Treatments

While lung diseases spawn from a multitude of factors, the way they manifest can be categorized into a few types as described in the previous sections. Most lung diseases have the common symptom of shortness of breath, and therefore most basic treatments attempt to alleviate this by opening up the respiratory airways. There are also specific treatments directed at certain types of respiratory diseases. This section will cover the broad treatment options and a few of the more specific treatment options.

Medicine

In the case of medicine, there is rarely a "one size fits all" solution. Different concentrations and combinations can have variable effects, especially when generalized to all people.

Inhaled bronchodilators are often a preferred treatment approach because they are delivered straight to the airways and lungs and work very quickly. They are often used to treat obstructive diseases like asthma and COPD due to their ability to relax the airway muscles, making it easier to breathe. There are different types of bronchodilators and the specific one used is dependent on the patient characteristics. Certain products are fast acting (albuterol),

while others provide a more lasting relief (formoterol, salmeterol, tiotropium). Inhaled corticosteroids can also be employed to treat airway inflammation.

If the cause is related to mucus buildup, expectorants, which help loosen the mucus in your lungs, can be prescribed. They often are combined with decongestants, which may provide extra relief. Mucus thinners, such as acetylcysteine, make mucus easier to cough up by loosening it.

If the respiratory disease is caused by ongoing inflammation, which can apply to both restrictive and obstructive diseases, oral medicines that suppress the immune system may be used. These include corticosteroids such as prednisone and immunosuppressants like azathioprine among others.

Infectious causes of lung diseases, such as bronchiectasis are managed by initiating prompt antibiotic therapy with oral antibiotics such as amoxicillin or macrolides and in more serious cases, intravenous antibiotics.

Medications available to treat most causes of restrictive lung disease are limited. Two drugs, Esbriet (pirfenidone) and Ofev (nintedanib), are FDA-approved to treat idiopathic pulmonary fibrosis. They act on multiple pathways that may be involved in the scarring of lung tissue. Studies show both medications slow disease progression in patients based on objective spirometry measures, although experts have yet to reach a consensus on their effectiveness. Other evidence shows the antioxidant N-acetylcysteine may help prevent lung damage in these patients.

While not typically thought of as a medicine, drinking plenty of fluid, especially water, helps prevent airway mucus from becoming thick and sticky. Good hydration also helps humidify the respiratory tract and keeps mucus moist and slippery, making it easier to cough up.

Oxygen Therapy

Many lung diseases result in low levels of oxygen in the blood due to poor air intake. Supplemental oxygen therapy aims to augment the oxygen supply and can help reduce shortness of breath. Depending on the case, oxygen therapy may only be needed during sleep and exercise, while in more severe cases it is needed on a continual basis. Non-invasive positive pressure ventilation (BiPAP) is also a commonly used method. It uses a tight-fitting mask and a pressure generator to assist breathing and is helpful for people with obesity hypoventilation syndrome and in patients with specific nerve or muscle conditions causing restrictive lung disease.

Lifestyle Changes

Many lung diseases spawn from poor lifestyle choices. The good news is, many of these habits can be changed with effort and there are pulmonary rehabilitation programs to help facilitate and assist patients with these positive lifestyle changes. Common lifestyle changes include, smoking cessation, healthier eating (especially if obesity is a cause), regular exercising, education, breathing therapy, and living and working in an environment with cleaner air.

Surgery

Surgery usually is a last resort for people who have severe symptoms that have not improved with first line approaches including medicines and/or adjusting lifestyle.

When the walls of the air sacs are destroyed as in COPD, larger air spaces called bullae form and can grow large enough to interfere with breathing. In a bullectomy, surgeons remove one or more very large bullae from the lungs. In lung volume reduction surgery (LVRS), surgeons remove damaged tissue from the lungs. In carefully selected patients, LVRS can improve breathing and quality of life.

The most extreme surgery is a lung transplant in which surgeons remove the damaged lung and replace it with a healthy donated lung. This is usually only recommended when the condition is quickly worsening or very severe. A transplant comes attached with many risks. New infections can emerge post transplant and the host body could reject the donor lung thinking the transplanted lung is a foreign threat. Furthermore, the supply of donor lungs is limited relative to the long waiting list of patients that could benefit from a lung transplant. There are specific criteria that attempt to allocate the limited and precious supply of donor organs in a fair way.

Conclusion

The aim of this section is to provide insight into the broad spectrum of lung diseases as well as the common causes and treatments. The next section, 3.2 will explore the severity, frequency and demographic distributions of the most prominent lung diseases.

3.2 Epidemiology

In order to emphasize the necessity for pervasive spirometry use, a brief survey of the epidemiology of respiratory diseases will be summarized. The terminology used to quantify magnitude and severity is shown in Table 3.2 Respiratory diseases are among the leading causes of death worldwide as shown in 2008 worldwide data in Table 3.3. Unlike many other causes of deaths, the mortality rate for respiratory diseases is actually rising with respect to time. Between 1980 and 2014, the rate of death from chronic respiratory diseases, such as

Tab. 3.2: Relevant terminology for the epidemiology section

Term	Definition
prevalence	The prevalence of a disease measures the number of cases of existing disease in the population at a given time, or over a period such as the past 12 months and is expressed as a percentage. It is calculated as the number of people with the disease divided by the total population, and is usually expressed as a percentage.
incidence	The incidence of a disease measures the number or rate of new cases of disease occurring in the population, over a specified period such as 12 months. Annual incidence is calculated as the number of new cases of a disease occurring in 12 months divided by the population who were disease-free at the beginning of the period (which can be hard to measure).

COPD, increased by nearly 35% overall in the US [19]. This rise peaked in 2002 and has since dropped by about 5%, most likely due to the widespread campaign surrounding the harmful health effects of smoking and the growing trend for smoking cessation.

Tab. 3.3: The 10 most common causes of death in 2008 [83]

Deaths attributed to	Worldwide
Ischaemic heart disease	7.3 million (12.8%)
Cerebrovascular disease	6.2 million (10.8%)
Lower respiratory infections	**3.5 million (6.1%)**
COPD	**3.3 million (5.8%)**
Diarrhoeal diseases	2.5 million (4.3%)
HIV/AIDS	1.8 million (3.1%)
Trachea/bronchus/lung cancer	**1.4 million (2.4%)**
Tuberculosis	**1.3 million (2.4%)**
Diabetes mellitus	1.3 million (2.2%)
Road traffic accidents	1.2 million (2.1%)

In this period, 85% of the deaths (3.9 million people) were from COPD, skyrocketing it up to the third leading cause of death in the US as of 2014. Other chronic respiratory illnesses that saw dramatic increases included: particle-inhalation diseases, such as pneumoconiosis and interstitial lung disease, asthma, and pulmonary sarcoidosis. [18]. Globally, Lung infections such as pneumonia or tuberculosis (TB), lung cancer and COPD together accounted over 10 million deaths worldwide in 2008, comprising of one sixth the global total. The World Health Organization (WHO) estimates the same four diseases accounted for one-tenth of the disability-adjusted life-years (DALYs) lost worldwide in 2008 as shown in 3.4 [83]. The impact of the most common respiratory diseases is illustrated in 3.5

Tab. 3.4: The 10 most common causes of disability-adjusted life-years (DALYs) lost worldwide in 2008 [83]

DALYs lost to	Worldwide
Lower respiratory infections	**79 million (5.4%)**
HIV/AIDS	65 million (4.4%)
Ischaemic heart disease	64 million (4.4%)
Diarrhoeal diseases	56 million (3.8%)
Cerebrovascular disease	48 million (3.3%)
Road traffic accidents	45 million (3.1%)
COPD	**33 million (2.3%)**
Tuberculosis	**29 million (2.0%)**
Diabetes mellitus	22 million (1.5%)
Trachea/bronchus/lung cancer	**13 million (0.9%)**

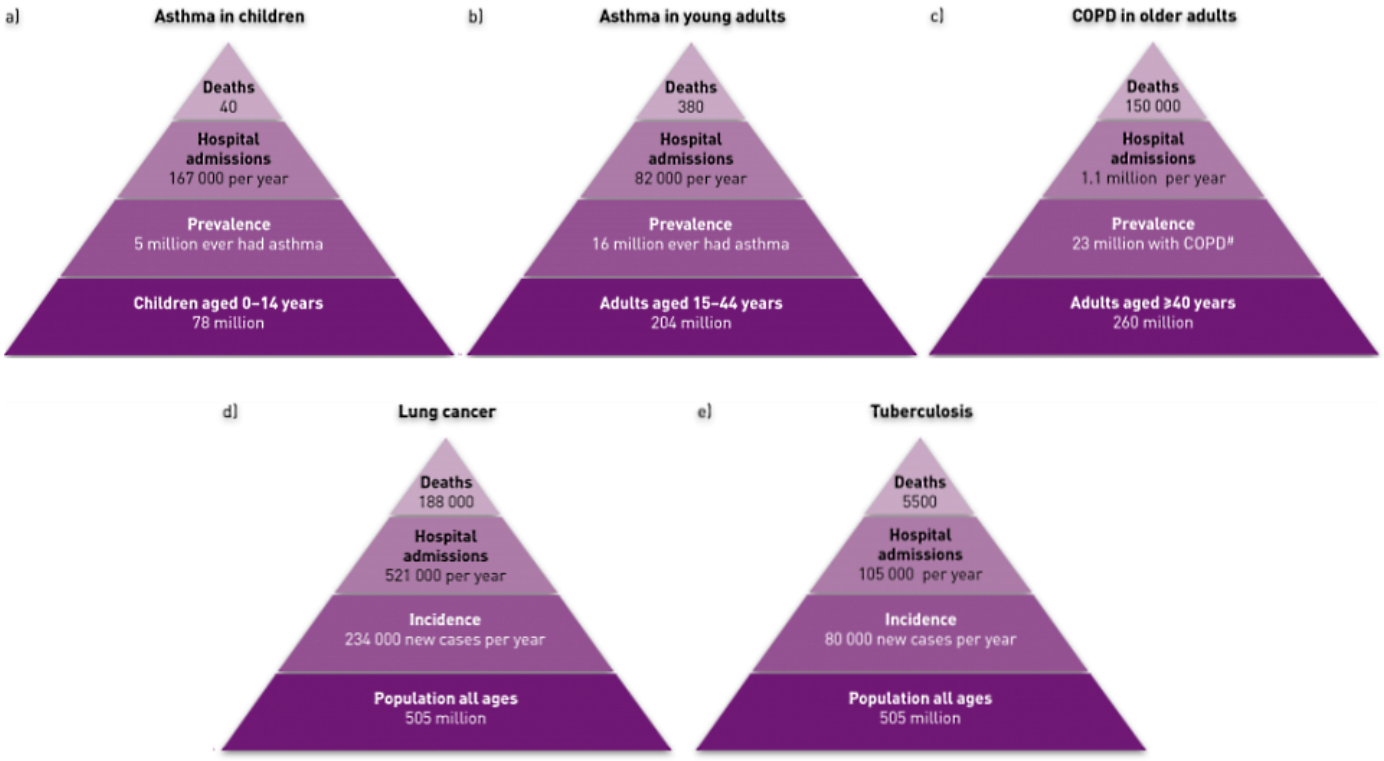


Fig. 3.5: The burden of various respiratory diseases, around 2010 [83].

Although asthma causes few deaths, it is a significant cause of disability, especially for children and young adults [68]. Furthermore, prevalence of childhood and adult asthma has increased in the last twenty years, and is now at its highest level. Asthma prevalence increased from 7.3% in 2001 to 8.4% in 2010. In 2010, an estimated 25.7 million people had asthma: 18.7 million adults aged 18 and over, and 7.0 million children aged 0–17 years [2].

By 2030, the WHO estimates the four major potentially fatal respiratory diseases (pneumonia, TB, lung cancer and COPD) will account for about one in five deaths worldwide, compared to one sixth of all deaths found globally in 2008. The magnitude is expected to remain stable at about one-tenth of all deaths, with an increase in COPD and lung cancer deaths and a decline in deaths from lower respiratory infections and TB.

Conclusion

Respiratory diseases are therefore likely to remain a major burden globally for decades to come. Prevention, diagnosis, tracking and treatment of lung diseases will need to be improved in order for the impact on longevity and quality of life of individuals, and their economic burden on society, are to be reduced worldwide. The focus of this work is in improving screening and tracking, although there are inherent aspects that can aid in prevention and treatment.

4 Spirometry Background

> *It's supposed to be hard. If it were easy, everyone would do it.*
>
> **— Tom Hanks**
> (A League of Their Own)

4.1 Spirometers

The trajectory of most device innovation and impact is contingent on the technology available at the time. There are perhaps four significant technology shifts that have occurred since the 20th century: mechanical, analog, digital and distributed. For example, scientific computing was once done by mechanical slide rules which were replaced by bulky analog computers which then became faster and smaller digital computers, which were more recently dethroned by today's state of the art; distributed cloud computing, which leverages hundreds of servers around the world. This trajectory is no different for spirometry, although it somewhat lags in comparison to computing. One of the goals of this work is to advance spirometry using the new tools made available with the advent of distributed, data-driven computing. First, a brief history of spirometry followed by a survey of modern devices will be discussed so the improvements enabled by this work can be fully appreciated.

4.1.1 History

The first recorded attempt to measure lung function dates back to the second century, when Galen, the famous Greek physician, tried to determine respiratory volume by having a child breathe into a bladder. Science has never claimed to be glamorous. For the next few thousand years, numerous other methods and inventions supporting basic lung function measurements came and went. In the 1840's John Hutchinson, an English surgeon invented a device he called a "spirometer" which is Latin for "breathe measure". The device was about the height of a person and was essentially a calibrated bucket inverted in the water. Lung volume could accurately be measured by simply instructing a patient to take a deep breathe and exhale into a tube connected to the bucket. Dr. Hutchinson also coined the term "vital

capacity" or in other words, capacity for life, because he observed this metric was predictive for premature mortality. He went on and pitched the device to life insurance providers as a legitimate method for predicting life expectancy, but it never caught on. Dr. Hutchinson did not give up easily. On a quest to give his invention credibility, Dr. Hutchinson evaluated about 2000 patients and began to notice a strong correlation between height, age, weight, and the volumetric vital capacity. This finding solidified the spirometer as a clinical device and the field of pulmonary function testing began to emerge. For the next 100 years, the design improved, but the basic functionality remained the same. A survey of these early spirometers, including Dr. Hutchinson's original design, is shown in Figure 4.1.

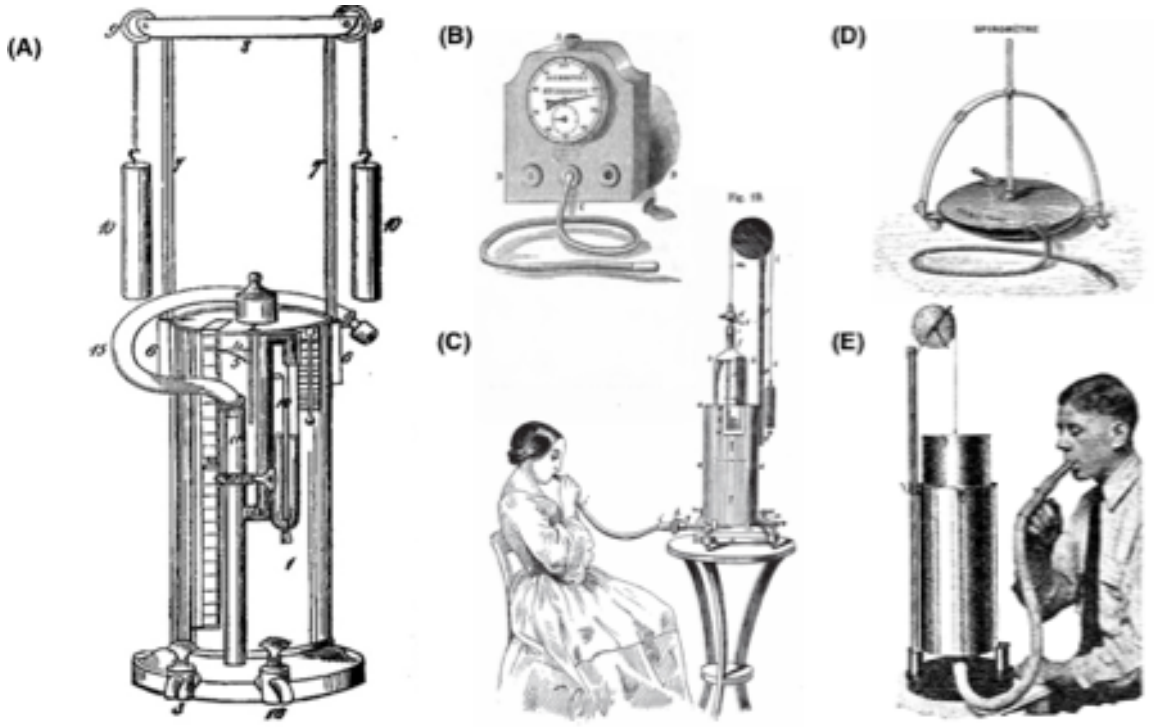


Fig. 4.1: A survey of early spirometer designs. (A) and (C) show Dr. Hutchinson's original spirometer design, (B) is Gardiner Brown's "spiroscope" from The Science and Practice of Medicine, (D) Boudin's spirometer design from 1854, later sold in 1905 (E) the Sanborn spirometer from 1925

In the mid 20th century, a paradigm shift occurred in the field when it was determined that 90% of the predominant respiratory disorders (COPD, asthma, etc) were obstructive (limited flow rate) in nature. In general, restrictive diseases can be diagnosed given total lung volume, but obstructive diseases require metrics based on flow rate. A published report by Stead and Wells determined the traditional water type spirometers were adequate for measuring lung volumes but were not suitable for accurately measuring flow rates [24]. The state of the art spirometer needed to accurately measure volume, and flow. In 1960 Jones Medical introduced the first non-water based spirometer which was easier, cheaper, more accurate, and more hygienic. Also around this time, Dr. Tiffeneau introduced a metric that measured the volume of air exhaled in a given time known as forced expiratory volume

(FEV). FEV later proved to be effective for quantifying obstructive disease severity and could be measured with flow capable spirometers such as the one made by Johns Medical. These innovative products and metrics marked the beginning of a new era where early detection of pulmonary disease could be done at a standard physician's office.

4.1.2 Modern Spirometers

Modern spirometers are generally flow based by design, directly measuring the instantaneous exhaled flow and deriving volume and other metrics. Since there are multiple avenues for measuring instantaneous air flow, modern spirometers can come in a few flavors. The core technologies are defined in Table 4.1. These technologies have inherent pros and cons. Pneumotach devices are very accurate, but extremely sensitive to temperature, humidity and altitude and thus require daily calibration and often other onboard sensors. Velocity based anemometers are not affected by environmental variations but are often less precise and experience degradation over time. Ultrasonic spirometers are newer to the market and require more precise, intricate manufacturing, but do not need regular calibration. Hot air anemometers, while simple, are somewhat of a lost cause as they do not measure airflow direction and require significant calibration with each use. Pneumotachs are the most prevalent spirometers in medical clinics because they have stood the test of time, are relatively cheap, and very accurate. Regardless of the design, the basic concept remains fixed: during a spirometry test, a patient exhales through a flow-monitoring device (typically a tube or mouthpiece), which measures instantaneous flow and cumulative exhaled volume. The details of spirometry testing are covered in the upcoming Spirometry chapter.

Tab. 4.1: Modern Flow Based Spirometry Technologies

Technology	Description
pneumotach	measures differential pressure measure across a membrane of known resistance based on the Venturi effect.
velocity aenometer	convert airflow into measurable rotational energy using a turbine design. Flow rate is proportional to turbine rotational velocity.
ultrasonic	measures flow by leveraging the Doppler effect, essentially measuring the time of flight of inaudible sound, which is sensitive to subtle variations in the air due to variable airflow.
hot wire aenometer	measures the electronic resistance through a hot wire which varies with the temperature of the wire. When airflow passes the wire, the temperature drops proportionally to the speed of the flow.

Most clinical grade spirometers cost over $1000 and can be as large as a refrigerator. The more portable variants are small enough to take home but are not designed to be portable.

There is also a growing market of cheaper options ranging from $20 to $500. These tend to be battery powered and much smaller, but typically only have a few simple metrics such as peak flow or total lung volume and are slowly gaining acceptance [12, 66]. See Figure 4.2 for examples of modern portable spirometers. There are also plastic mechanical peak flow meters less than $20, but peak flow is generally considered to be a poor indicator of lung function and is therefore not preferred [70].

Fig. 4.2: A sampling of modern portable spiromtery products. The products shown are (A) Microlab by Vyaire, (B) Spirodoc by Amplivox, (C) Datospir Micro by Sibelmed and (D) EasyOne by Nddmed.

4.1.3 Portable Spirometers

Spirometers advertised as portable are often missing several vital functions required for them to be considered truly portable in the 21st century. For example, many have no accessible internal storage requiring patients or clinicians to record the data by hand or require a laptop for operation. The optimal criteria for a modern portable spirometer is as follows:

- A rechargeable, long-lasting battery
- Wifi, NFC or Bluetooth upload capabilities
- Offline retrievable storage
- Web or app interface for reporting history and trends
- Fits in a purse or small bag
- Disposable or easy to clean mouthpiece
- Durable, dustproof
- Less than $500

Perhaps the easiest way to meet all of these requirements is to leverage technology that innately meets several of them; a smartphone. There are a growing number of new, innovative products that utilize a smartphone for computation, display, and storage while employing a small low powered handheld device for capturing the airflow and transmitting

the data to the smartphone wirelessly. Using a smartphone for the heavy lifting is beneficial in many ways. First off, it is far cheaper for the end user assuming they already own a smartphone and costs are less for the manufacturer as they do not need to include the display and powerful CPU components in the core product. Phone manufacturers have already optimized their smartphones for speed, portability, and longevity, and additionally created a vast network of documentation and support tools for developing professional grade applications on the platform. The smartphone industry has doubled in revenue to nearly 500 billion dollars since 2013 and the tech inside is evolving even faster [25]. This powerful foundation allows researchers and developers working on spirometry products to focus on the core external device that interfaces with the phone, rather than all components for a stand-alone spirometry product.

Development of an accurate, portable, sanitary, long lasting spirometer can be done separately from the mobile app which must receive the data then process, display and upload the results. Furthermore, the clinical certification process is focused on the accuracy of the measurement only and therefore is only concerned with the external flow meter. Once the meter is certified, developers have the freedom to improve the app experience and try out different interfaces, without needing to go through the labor intense process of re-certification. At the end of the day, everyone benefits from this iterative development style. The users do not need to purchase new hardware when the interface is updated, they simply download an updated application. Developers have the freedom to try new ideas and get them into the field without the certification speed bumps. Two of the most promising smartphone based accessory spirometry products are shown in Figure 4.3. These products provide all of the metrics traditional spirometers measure but are also cognizant of the user's needs as shown in the simplistic, pleasing design of the core hardware and app. These products are still in infancy stages of development and have a lot of ground to cover before they can render traditional clinical products extinct. Nonetheless, they have come a long way progressing from the academic echo chamber into the world of consumer products in less than a decade [29].

One of the largest detractors preventing widespread adoption of these products is the need for external hardware. For spirometry to truly be ubiquitous, it must be fully integrated into products consumers already have, such as "smart" phones, hubs, or watches. A number of applications (e.g., Spirodroid, mySpirometer, Spirometer Pro, Smart Peak Flow) exist in the iPhone and Android stores, but many of them are inaccurate, hard to use (see reviews), have only been tested on a small population and have no form of clinical validation or medical assessment. An ideal spirometry app must be convenient, usable, regulated, and most importantly, trusted by physicians.

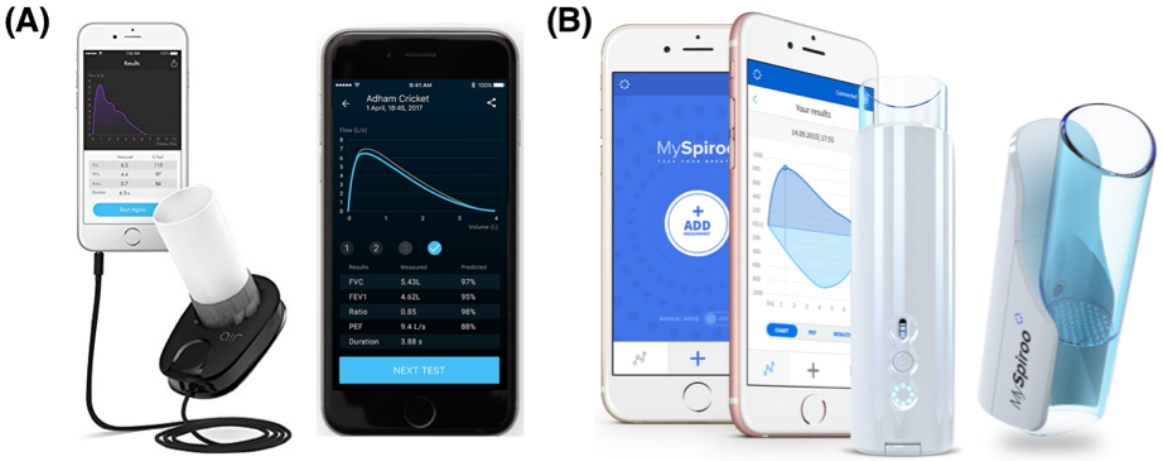


Fig. 4.3: The most promising smartphone-based portable spirometry products. (A) the Nuvoair spirometer and app which connects to a phone via wire and can be purchased for $250. (B) the Myspiroo is the most premium, high tech option as it is wireless and supports Bluetooth and NFC and has many other supplemental sensors that many clinical spirometers are missing, such as humidity sensors. It is not yet available for sale, although the app can be downloaded.

The Ford Model T became known as the "freedom machine" of the 20th century because it liberated Americans from the confines of their town enabling the widespread flow of products and ideas on a national level. While the Model T revolutionized the transportation of physical goods for the average consumer, smartphones have the same effect in the digital information realm, but on a global level. Smartphones have become the standard in communication, navigation, organization, and information and have brought a wave of disruption in decades-old industries from photography to taxi cabs. It is only a matter of time before this wave impacts the personal care and health industries. In the next section, the emerging field of smartphone-based mobile health will be outlined to show it is indeed the future of personal health and therefore a compelling platform for pervasive spirometry.

4.2 Mobile Health

Mobile health, sometimes referred to as mHealth, has exploded in the last few years. Apps and wearables that monitor and track fitness, sleep and nutrition have become highly sought after for a broad spectrum of consumers. In fact, health and fitness apps have grown by over 300% in the last three years, according to a recent study and globally, there are over a billion people who own smartphones [21]. This growth does not come as a surprise. The pursuit of knowledge is baked into human nature such that anything that provides answers to difficult questions is blindly revered. It is because of this intrinsic quality that religious deities, search engines and more recently health tracking have mass appeal. People want

to know they and their loved ones are going to be alright and do not want to wait for their physician to reassure them.

Despite the massive growth and customer satisfaction, most of these products have little to no clinical merit and tend to tackle the low hanging fruit in the health and wellness space. For example, a number of sleep tracking products rely on motion, sound and sometimes heart rate to predict sleep state and quality. While this is much more convenient and comfortable than the traditional clinical approach which requires sleeping in a lab with invasive electrodes and sensors, it has little to no clinical significance. Recent studies have shown when dozens of similar sleep products are compared in a controlled environment, the results differ significantly [94]. In the clinical world, this means they do not work. For a predictive instrument to be considered clinical, it must achieve a level of accuracy comparable to the current standard, and the predictive capabilities must demonstrate a high level of reproducibility when utilized within the operational bounds. With the exception of step counting and wearable based heart rate monitoring, most mHealth solutions are far from being considered clinical as they lack the needed precision and reproducibility or have not been thoroughly vetted through clinical trials and certifications. Nevertheless, existing mHealth applications provide strong evidence to the massive demand for mobile-based sensing. This demand will no doubt motivate researchers, clinicians and tech companies to collaborate and establish the much-needed clinical utility for these applications.

4.2.1 Liberating Spirometry

Smartphone-based mHealth is the James Bond of the health industry. Agent Bond constantly finds himself up against wealthy, powerful villains, politicians and businessmen. He constantly must overcome seemingly impossible circumstances, sometimes by jumping out of an exploding building, and other times mentally outsmarting his deceitful foes. Despite the treacherous, emotional journey, by the end of every movie, James Bond inevitably prevails as any other outcome would defy the expectation of the viewers. mHealth may not have to withstand hours of torture or drive motorcycles off cliffs, but for widespread adoption, it must navigate through a minefield of challenges set in place by clinical standards as well as the expectations of modern consumers. This section will set the stage for a mHealth based spirometry product by outlining the business case, potential challenges, as well as the inherent benefits of widespread mobile spirometry.

The Customer is Always Right

The age-old phrase, "the customer is always right" highlights the parasitic relationship between a business and a customer. For a business to survive it must pivot and adapt based on customer demands and feedback. This is why services like next day shipping or unlimited warranties exist. As it turns out, people are relatively simple and adhere to many of the basic

laws of physics and nature. When presented with multiple paths, they take the path of least resistance, optimizing for long-term survival first and instant gratification second. Unlike the traditional consumer-business relationship, the patient-doctor relationship is symbiotic; they need each other to survive. Because traditional spirometry has a defined outcome on survival, many people are willing to spend the time and money on frequent checkups despite the inconveniences. As soon as viable alternatives to visiting a clinic emerge, the patient now has options and must be treated more like a consumer who will always take a path of least resistance. Therefore, if mobile spirometry yields similar health outcomes to traditional clinical based approaches, it will no doubt become the preferred option for patients.

4.2.2 Inevitable Challenges

Spirometry is unlike other typical clinical metrics such as pulse or body weight in that the clinicians who administer tests must be highly trained in order to coach patients through the unintuitive maneuver to ensure the results are accurate and meaningful. This necessary insurance adds cost to the procedure and limits its use to the physician's office. Often times the standard 14-hour training for spirometry is inadequate or non-translatable to reality. In a 2010 study, following the standard training clinicians were assessed for adherence to American Thoracic Society (ATS). The study showed that after nine months clinicians could only produce acceptable spirometry testing in 60% of their patients [10]. Why is it so hard to get acceptable results? There are several potential sources of measurement error in spirometry which are outlined later in Chapter 5. Many of these errors such as slow start or variable flow are difficult for a clinician to spot because of they are a byproduct of how the patient exhales into the instrument and are therefore not clearly visible or audible. These errors can often be identified when analyzing the results, which suggests the process of post-effort error identification could be automated. In fact, prior work has demonstrated automatic spirometry error characterization is possible and comparable to human level performance [54]. Furthermore, unless the clinician is highly experienced, coaching tends to be dispassionate and relatively generic similar to the mandatory seatbelt instructions provided by flight attendants. A similar outcome could be achieved with a short instructional video. So far these insights are promising for the transition from clinical grounded spirometry, but crucial challenges remain undressed.

In order for a physician to excel at their job, medical records must be organized, filtered and presented in an efficient, unbiased, and consistent way. Currently, a standardized, trusted method for uploading and storing health data does not exist, although researchers are bringing several new ideas to the table from intuitive user interfaces to distributed blockchain databases [45, 60]. It is therefore crucial for researchers, developers, and care providers to collaborate in order to reach a viable solution that satisfies the needs of both patients and physicians.

In summary, for spirometry to be effective outside of the clinic, the following requirements must be met:

- Automated coaching and feedback
- Automated maneuver error identification and quality assessment
- Automated upload and private storage of data
- Physician approved results and analytics

4.2.3 Benefits of Mobile Spirometry

The challenges presented above are difficult, but certainly not impossible given today's advances in artificial intelligence and sensing. In fact, novel solutions to these challenges are explored later in the Methods chapter and some already existing solutions are outlined in Related Work. The most compelling case for mobile spirometry does not come from the business case or improvements over the traditional methods, it stems from the intrinsic benefits that enable the use of spirometry whenever and wherever a patient desires.

Convenience

As mentioned earlier, people tend to be lazy. Services like pizza delivery, Amazon Prime and Netflix show, in general, people are willing to pay a premium to remain in the comfort of their home. In fact, there are few services remaining that do not have a model where the provider comes to you. Ironically, this is how personal care used to be. Around a century ago it was common for the local physician to come to the patient's home for a "house call", but this turned out difficult to scale as populations skyrocketed and families moved to suburban neighborhoods. As a result, we have a centralized system of hospitals that is fairly efficient given the complexity and available resources. There are, however, a few issues with the centralized model of care. First, given the scale of the operations, patients must be prioritized based on severity and economic status. For most people, this means they have to wait for care, hence the term "patient". A decentralized mHealth model will help reduce the load on hospitals by helping care providers prioritize patients in a more efficient way while also enabling patients to get instantaneous feedback and advice, perhaps making the term "patient" a thing of the past.

Comfort

Hospitals are stressful. The term, "white coat hypertension" describes a condition in which blood pressure spikes purely due to the stress induced by the clinical environment. It is no surprise over 30% of patients are affected by this form of anxiety which causes their body to enter an involuntary fight or flight state. A 2009 study found nearly 20% of Medicare patients who are discharged from a hospital develop an acute medical problem within the following 30 days that necessitates another hospitalization, often unrelated to the original

diagnosis [22]. While the readmission reason varies, it often comes down to either clinic induced stress or an acquired infection. Either way, this is a compelling reason to offset a portion of the traditional clinical operations with comparable mHealth counterparts.

Economic

Spirometry mHealth solutions offer significant cost savings for both the patient and care provider. COPD alone incurs a 50 billion dollar cost burden with an estimated prevalence of $4000 per patient per year [28]. More compelling statistics are reviewed in the Epidemiology section of chapter 3.

Mobile spirometry will not completely abolish this cost, but it will certainly put a dent in it. Many patients must visit the clinic multiple times per week to check their respiratory health and ensure their treatment is working adequately. These types of visits can certainly be outsourced to a mHealth based measurement. Insurance companies have already hopped on the mHealth bandwagon, rewarding customers with reduced premiums for achieving a step count above the desired threshold. Anything that reduces the risk of hospitalization is attractive to insurance companies and mobile spirometry certainly has this potential. Removing the requirement for expensive hardware and constant clinic visits is the most straightforward way to make spirometry inexpensive and accessible to a broader population, especially when the goal is to track or monitor treatment and symptoms.

Impact

In addition to cost savings and relieving patient stress and clinical burden, mobile spirometry has the potential for a massive positive impact on health outcomes. Currently, spirometry data is collected at a very low frequency since it requires the patient to visit a clinic or in some cases return home for each reading. These measurements, at best, are updated on a twice daily basis which can reveal macro trends if given enough time, but they may not be timed to accurately capture spontaneous exacerbations. Mobile spirometry enables patients to measure lung function at any point in the day and has the potential of exposing micro trends that can only be revealed with sufficient contextual data points. Measurements could be easily logged after a meal, during a hike, before bed, an hour after ingesting medication or pre/post asthma attack. This high-resolution information can enable both the care provider and patient to better understand the cause and effect pattern of their respiratory condition, allowing more targeted treatment options.

Many life-threatening asthma or COPD exacerbations are not spontaneous and instead occur following hours or days of lung degradation. Several studies have shown that daily spirometry measurements can accurately predict impending asthmatic episodes [36, 40, 12]. Such predictive trends empower the patient allowing them to stay ahead of their condition rather than at the mercy of it. With higher resolution data, these predictive trends will only become more powerful and insightful. With these trends, a mobile spirometry app could

provide insights on the collective information from several sources. For example, a local news source may report on a wildfire which triggers the spirometry app to warn the user of the impending degradation in air quality and provide mitigation suggestions for avoiding an asthma attack. Providing the patient with the power to stay ahead of their condition will not only improve their health outcome but allow them to feel in control of their condition, thus boosting morale.

The low cost and high utility of mobile spirometry make it a tool for everybody, not just patients with a prior respiratory condition history. This is especially true if the tool is in the form of an app with no extra hardware, requiring little to no commitment. People trying to quit smoking could use mobile spirometry to somewhat gamify the process by taking several measurements throughout the weeks of recovery, using the steady improvement in lung function as a motivator to stick with the program. Spouses with snoring partners could encourage the use of mobile spirometry as a predictor for snoring severity and perhaps identify the conditions necessary for minimizing the issue and preventing it from ballooning into a more serious form of sleep apnea.

Integrating spirometry into a mobile platform allows sharing of the inherent benefits with the connected world. Coupled with smart home sensors that measure environmental conditions like air quality, temperature, and humidity, insightful correlations may be unveiled which can influence positive household changes that improve health outcomes for the whole family. Imagine a home that learns and adjusts for the ideal environmental state for each bedroom such that respiratory health, sleep and overall comfort are optimized for. Additionally, mobile spirometry can be used within the context of other health and fitness apps. Sleep quality apps could recommend mobile spirometry if sleep quality is dwindling. Fitness apps could include spirometry in addition to weight and heart rate for post workout analytics. More data allows for more insights which in turn motivates informed treatment interventions and positive changes to daily lifestyle.

The benefits of mobile spirometry are not limited to developed regions where smartphones exist in the pockets of most citizens. Smartphones set up for clinical-only use can be deployed to developing countries where traditional spirometry is nonexistent due to the cost and required expertise. Large populations can be efficiently screened via mobile spirometry by a handful of dedicated clinicians and individuals identified as high risk can be prioritized for further testing and treatment. This concept can be extended to other mHealth solutions outline later in Related Work empowering a single device to be used as a multipurpose too for health screening.

This all being said, for these exciting use cases to be considered valuable and credible, the spirometry measurement must be subject to a high level of scrutiny and must be validated

using existing clinical standards, otherwise, the utility is closer to that of a game rather than a medical device.

4.3 Conclusion

In this chapter, the history of spirometry, as well as the modern variants, are outlined. It is clear that what is considered modern in the world of spirometry is viewed as antiquated to other industries and general consumers. This observation is less of a stab at spirometry and more of a reflection on the moribund medical device industry who's technical ineptness can be quantified by the number of floppy disks actively being used. Whether the transformation of the medical industry is devastatingly rapid like fall of taxis or more likely, cautiously slow like the adoption of autonomous vehicles, it is nonetheless inevitable. The demand for such a shift is supported by advances in technology including the massive growth in the mHealth space and ever-increasing decentralization of other similar industries. There are, however, obvious risks exclusive to the health industry so it is imperative that modern replacements to medical devices are rigorously tested and validated with caution. Fortunately for spirometry, transcendence into the mHealth world is backed with over a century of comprehensive clinical guidelines and rules. As the next chapter will show, these well-established rules equip clinicians with a powerful swiss army knife for diagnosing respiratory conditions and they are derived from a single signal that can be measured externally: airflow.

5 Spirometry

> *We are all in the gutter, but some of us are looking at the stars.*
>
> — **Oscar Wilde**

The field of astronomy exists because astronomers have measurement tools and theories that enable a wealth of knowledge to be extracted simply from the light emitted by distant stars. Spirometry has similar potential as it enables crucial properties of the respiratory system to be derived from exploiting and measuring its most external signal: airflow. As a result, spirometry is the most common lung function test and is preferred because it is relatively non-invasive and provides a repeatable snapshot of general lung health. Many lung diseases such as COPD, asthma, pulmonary fibrosis, and cystic fibrosis directly or indirectly alter the speed at which air travels in or out of the respiratory system. This is exactly what spirometry measures, which makes it a great tool for the diagnosis and monitoring of these respiratory diseases.

Spirometry maneuvers test the limits of lung function because monitoring normal lung function often fails reveal anything until degradation is severe. The common forced expiratory spirometry maneuver is designed to highlight the decline in function by forcing the patient to expel the maximal volume possible with maximal effort. This allows physicians to detect the early onset of both restrictive and obstructive diseases.

This section will cover the details of performing and interpreting a spirometry test, with a focus on the forced expiratory maneuver. The following information is based on the clinical standards defined by the American Thoracic Society (ATS), National Heart, Lung, and Blood Institute (NHLBI) and other well-accepted clinical guidelines [35, 79].

5.1 Procedure

Before the testing procedure is started, standard personal information is required such as age, weight, height, and ethnicity. This info is required to compute predicted normal lung function parameters which are then compared to actual measured parameters to detect

deviations from what is considered "normal" for similar populations. Next, the clinician educates the patient on the procedure, coaches them on the maneuver and makes sure they are prepared for the test. In the forced expiratory maneuver, the patient is instructed to inhale as much as possible and then exhale into the spirometer as hard as possible for as long as possible, directing all flow into the mouthpiece. If forced inspiration is also being measured, they are instructed to take a rapid deep breath after the forced exhale. The patient is provided a sanitary, disposable mouthpiece and in some cases is instructed to wear to nose clip to ensure no air escapes through the nasal passages. The forced expiratory maneuver is commonly used as it captures the limits of forced flow rate as well as the forced vital capacity. During the forced expiratory maneuver, the spirometer monitors the flow of air versus time and from this derives several predictive respiratory health metrics and curves. A typical session involves three or more trials and the clinician must ensure enough trials demonstrate reproducibility before recording the final results. The spirometry metrics and reproducibility criteria are covered later in this chapter in Section 5.4. The results are often available shortly after the session ends, although other pulmonary function tests may be required before a diagnosis is given.

5.1.1 Risks

Spirometry is associated with little risk; however, forceful exhaling can increase the pressure in your chest, abdomen, and eyes. For this reason, spirometry is discouraged or used with caution in patients who:

- Are pregnant or a small child /infant
- Have unstable angina
- Have had a recent pneumothorax
- Have had a recent heart attack or stroke
- Have had a recent eye or abdominal surgery
- Have coughed up blood recently and the cause is not known

5.2 Results

This section will cover what metrics are included in spirometry results, derived from the flow versus time sequence, as well as how to interpret them and diagnose lung conditions.

5.2.1 Common Parameters

There are several metrics that can be derived from the direct flow versus time signal measured by a spirometer. The most common metrics are shown in Table 5.1.

Tab. 5.1: Spirometry Metrics

Metric	Description
Forced vital capacity	**FVC**: The total volume of air expelled during the expiration
Forced expiratory volume	**FEV1**: The volume of air expelled in the first N seconds of expiration. Usually FEV1, where N=1 second is used.
Forced expiratory time	**FET**: The total time it takes to expire FVC
Peak expiratory flow	**PEF**: Maximum expiratory flow rate reached during the test
FEV1/FVC Ratio	The ratio of the FEV1 to the FVC, indicates percent of lung capacity expelled in the first second.
Percent predicted ratio	The ratio of the measured metric to the "normal" predicted value

5.2.2 Interpretation

Using the metrics outlined in the last section, coupled with the wealth of prior statistical and pathological respiratory health information, physicians can compile a comprehensive assessment of lung function from a single spirometry session. Spirometry results usually reveal one of four main patterns:

- Normal
- Obstructive
- Restrictive
- Hybrid obstructive/restrictive

Each of the four outcomes is described as followed [7]:

Normal

Normal readings vary depending on age, size, sex and ethnicity as these variables affect the size and age of the lungs. These patient-specific variables have a significant effect on predicted lung health as illustrated in Figure 5.1. Usually, a standardized reference table or algorithm is used with these personal variables to retrieve the predicted normal

readings [20]. Prior health statistics show that lung health deteriorates linearly with age, and intuitively, lung size increases linearly with height. Gender and ethnicity have been shown to affect chest and lung size independently of height. Weight is typically proportional to height and therefore somewhat redundant, but in cases such as obesity, the weight is no longer correlated to height and therefore effects lung function.

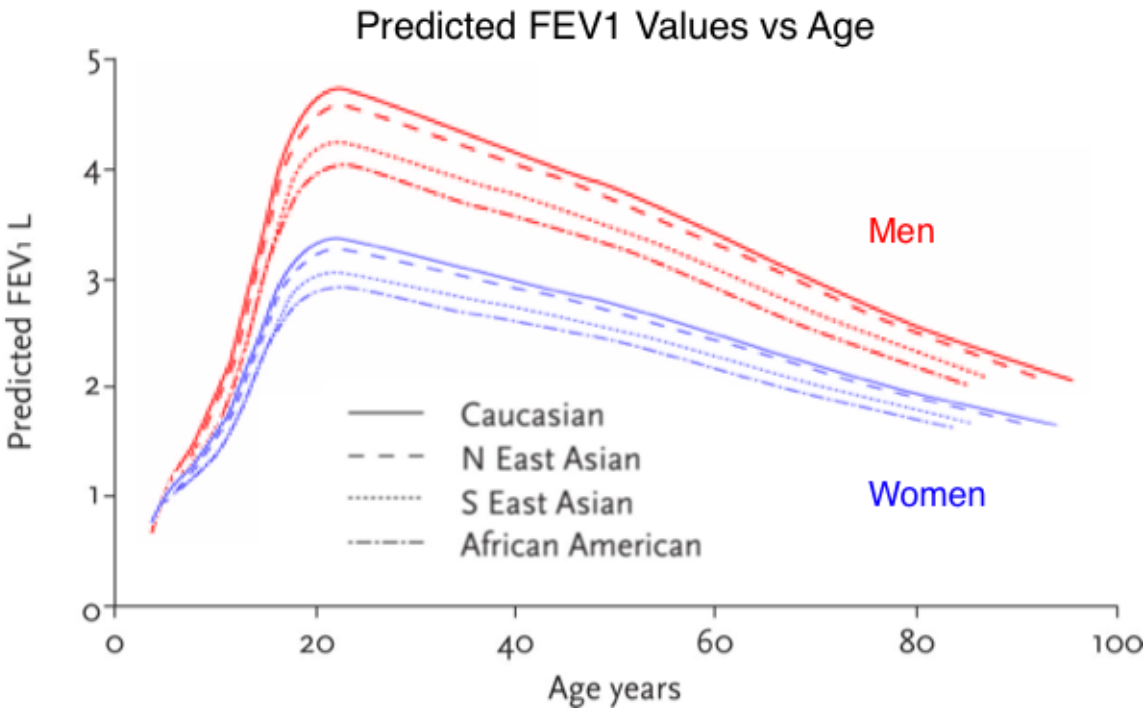


Fig. 5.1: Shows the age dependence for predicted FEV1 and the relationship for different genders and ethnicities

The FEV1/FVC ratio is a useful metric for measuring normalcy as it is independent of a patient's lung size. Prior research has found healthy patients can typically achieve an FEV1/FVC ratio of 0.7 or higher, which means at least 70% of the vital capacity was exhaled in the first explosive second of expiration.

A healthy individual's lung function measures are generally at least 80% of the values predicted based on their age, height, and gender [42]. Abnormal values of FEV1% are expressed as a percent of the predicted value and are typically categorized as follows [61]:

- Healthy: 80% or above
- Mild to Moderate Lung Dysfunction: 60-79%
- Moderate Lung Dysfunction: 40-59%
- Severe Lung Dysfunction: below 40%

There are exceptions to these rules of thumb, so it is generally preferred to completely rule out the other three abnormal patterns first in order to diagnose normality. Historical patient information, when available, should be used in place of predicted values as it is more applicable. For example, if a patient historically achieved 120% predicted FEV1 and a year later is measured at 100% predicted FEV1, there could be a serious issue in place despite an apparent classification of "healthy" based on calculated predicted values. However, since historical lung function data for patients are rarely available, predicted values are typically relied on.

Obstructive

As covered in Chapter 3, obstructive diseases result in narrowed airways and are usually caused by asthma and COPD. Narrowed airways hardly change the total lung volume, but have an observable effect on the rate at which air flows out of the lungs. Therefore, an obstructive pattern is characterized by a normal or slightly reduced FVC and a significantly reduced FEV1 when compared to predicted normal values. This also results in a reduced FEV1/FVC ratio.

For example, if a patient has an FVC above 80%, but an FEV1 below 80% the predicted value, then they likely have some degree of obstruction. It is also common to additionally check FEV1/FVC ratio, which if below 0.7, is an indicator of marked narrowing of the airways and obstructed airflow.

Restrictive

Restrictive patterns are usually caused by conditions that affect the lung tissue itself or affect the capacity of the lungs to expand and hold a normal amount of air. These conditions are typically a result of scarring or an external force preventing normal lung expansion. As a result, restriction is highly characterized by a reduction in FVC. The FEV1 is also proportionally reduced, so the FEV1/FVC ratio often remains normal. Therefore, a patient likely has restriction if the FVC is below 80% and the characteristics of obstruction are not present.

Hybrid

It is rarely the case in pathology that a condition or disease is clearly defined as black and white. Subsequently, the threshold-based analysis described above does not always provide a clearcut, concrete diagnosis. This can happen when a patient has mixed conditions, such as asthma plus another lung disorder. Additionally, there are lung conditions like cystic fibrosis that manifest with both obstructive and restrictive patterns where there is mucus buildup in the airways which results in narrowed airways, and restrictive scaring to the lung tissue.

In these cases, relying on the spirometry metrics may not paint the entire picture needed for a complete diagnosis. As a result, different tests may be used to provide additional insight to bring the diagnosis more clearly into focus.

5.2.3 Spirometry Curves

While the individual metrics introduced above provide a useful framework for assessing lung function in the majority of cases, they are merely a summary of the information embedded in the original airflow versus time curve and lack the information necessary for identifying errors in a trial or inspecting repeatability between trials. For this reason, it is common practice to also interpret flow versus time (FT) dynamics and the other derived curves described in Table 5.2. Most of the relevant information can be extracted from the flow versus volume (FV) curve, so it will be the focus of this section.

Tab. 5.2: Spirometry Curves

Curve	Description
Flow vs Time (FT)	The direct flow versus time as measured by the spirometer, typically not used as the other curves provide more relevant information and are preferred by physicians
Volume vs Time (VT)	The cumulative volume of air exhaled as a function of time. Computed by integrating the FT curve
Flow vs Volume (FV)	The result of plotting flow as a function of cumulative volume instead of time. Clearly shows the relationship of flow and volume in one plot

Normal VT and FV curves with the common metrics indicated are shown in Figures5.2. At the start of the test, both flow and volume are equal to zero. After the maneuver is started, the curve rapidly mounts to a peak, then descends at a rate proportional to the airflow speed. A normal, non-pathological FV curve descends in a straight or a convex line from top (PEF) to bottom (FVC) referred to as the expiratory curve. Similarly, in the cumulative VT curve, the volume rapidly increases during the explosive phase then steadily flattens as the final 20% of the vital capacity is exhaled and the FVC is reached.

Like spirometry metrics, curves can be used to detect obstructive and restrictive patterns. Figure 5.3 shows common examples of FV curve profiles compared to normal. Obstructive curves are characterized by a concave or "scooped" shape following the PEF in the FV curves while restrictive FV curves typically have a similar shape to their normal counterparts, but with a dwarfed scale volume-wise. Moreover, several other specific or hybrid conditions can also be discovered from the shape of the curves.

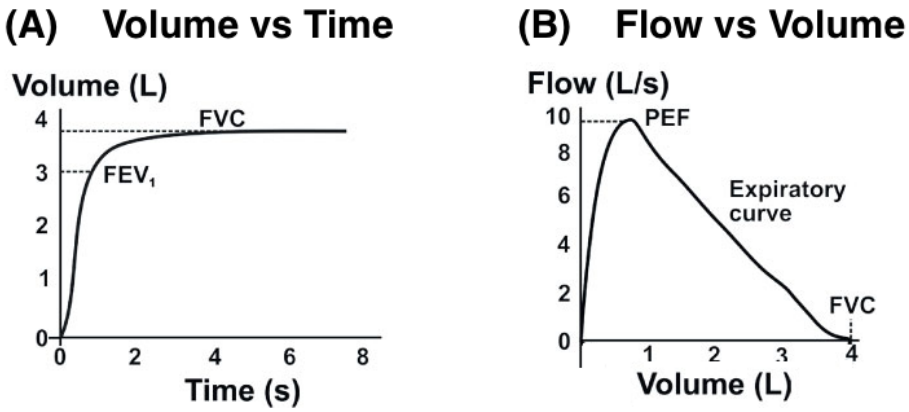


Fig. 5.2: Standard volume vs time and flow vs volume curves along with the commonly derived metrics, defined in 5.1

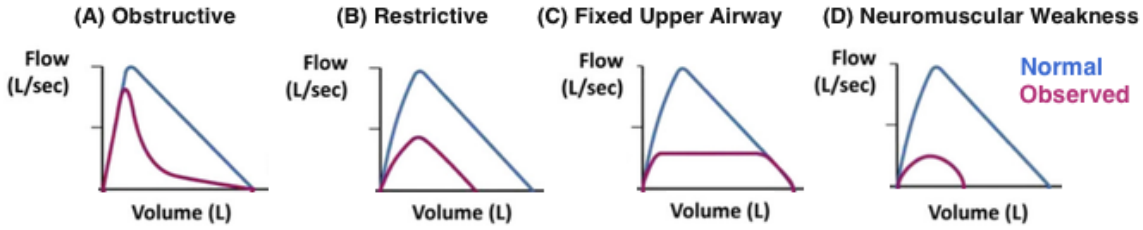


Fig. 5.3: Common conditions represented by their FV curve compared to the baseline normal curve

Curve Physiology

It can be useful to understand what is going on inside of the respiratory system during a forced expiratory maneuver and how it translates to the resulting FV curve.

The start of the maneuver is abrupt and explosive, quickly relinquishing the majority of the volume in less than a second until the PEF is reached. In this period, the large airways conduct the majority of the airflow and the shape of this region, as well as the magnitude of the PEF, are vastly dependent on the voluntary effort and muscular strength of the patient. As a result, PEF is considered effort-dependent and a poor indication of lung health since it is more based off of the patient's technique and strength.

The most import component of the FV curve is the decay from the PEF to FVC. This stage is considered effort-independent as it remains constant regardless of the patient's effort or strength. By this point, the majority of the air has expelled through the large airways and what remains must escape through the small airways. These small airways are the key to diagnosing obstructive conditions as they are often the most vulnerable airways in terms of obstruction.

The remaining region of interest is the final point on an FV chart or the max value of the VT curve. This point occurs when the flow has returned to zero or the volume has reached the FVC limit. The time at which this occurs is referred to as the forced expiratory time (FET) and signifies the end of the forced exhale. With or without obstruction, this point should be relatively constant as all of the air eventually is exhaled (except for trapped residual air which is covered in 5.3). In the case of restriction, this point tends towards zero as the severity of restriction rises.

5.2.4 Diagnosis Decision Tree

The diagnosis decision tree is shown in Figure 5.4 and adequately summarizes the clinical diagnostic process outlined in this section. Note that for a confirmed diagnosis in many cases, a total lung capacity (TLC) test, which is covered in the next section, is required. The TLC assessment cannot be determined with a spirometer. Nevertheless, a spirometer provides a wealth of preliminary knowledge, plus it is a simple and non-invasive tool. There are many variations of this tree as the communities have not seemed to converge on a single set of rules.

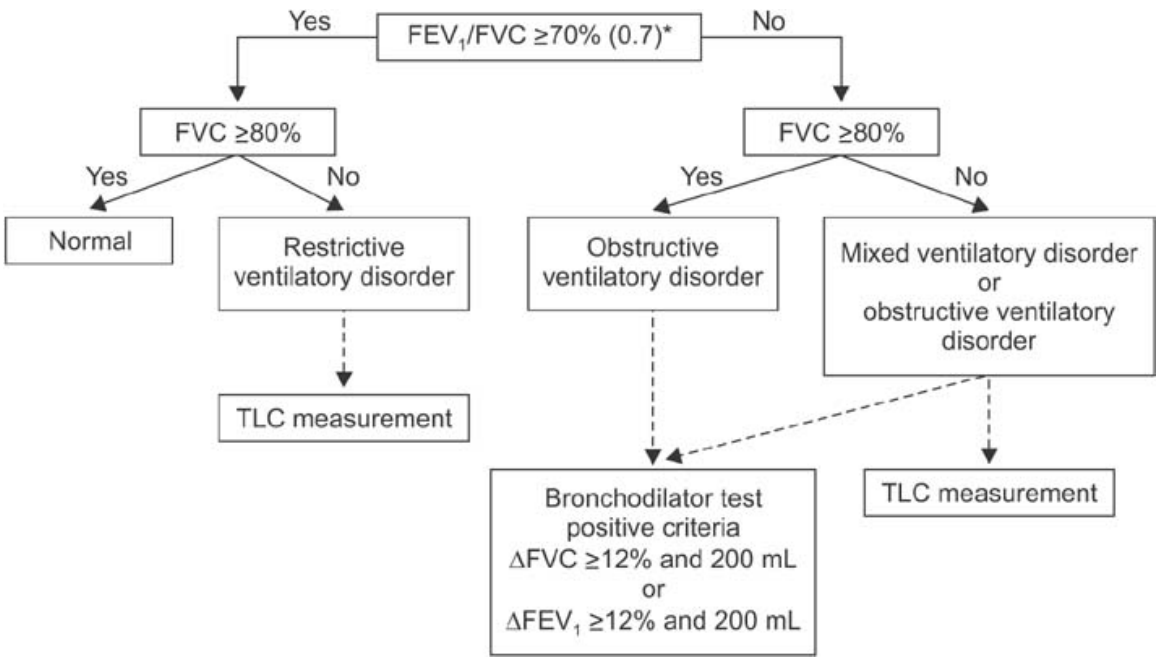


Fig. 5.4: A decision tree used for interpreting spirometry results and motivating follow up tests requried for full diagnosis [7]

5.3 Other Pulmonary Functions Tests

A spirometer can be used for tests other than what has been outlined so far. These additional tests are often employed to provide extra context so a complete diagnosis can be reached. Despite these additional tests, there are fundamental limits to what spirometry can measure before other, less convenient, instrumentation are utilized. The following are examples of additional tests that can be performed with a spirometer:

Reversibility Testing (bronchodilator)
Spirometry can also help to assess if treatment, such as inhalers, can effectively open up the airways. Theoretically, spirometry readings will improve if the narrowed airways become wider after administration of the medication. This process is called reversibility. Reversibility can be employed to quantify the severity of obstruction and is also useful in cases where the diagnosis of the lung condition is not clear. A spirometry reversibility test usually follows the following steps:

A baseline standard spirometry session is performed, followed by the use of an inhaled fast-acting bronchodilator (often Salbutamol). After roughly fifteen minutes, a repeat spirometry session is performed and referred to as the post-bronchodilator result. In the case of asthma, which is considered highly reversible, a "significant" improvement in FEV1 will typically be seen after administering the bronchodilator. If FEV1 does not increase "significantly" following the bronchodilator, the obstruction is more likely to be caused by another pathology, such as COPD and other tests will be necessary to make a definite diagnosis. The ATS defines a "significant rise" as a post-bronchodilator rise of at least 12% and by 200mL [61].

It is typical for COPD to be graded according to severity, in terms of the FEV1 measurement after a bronchodilator medication has been given to open up the airways. As a guide, the following post-bronchodilator values help to diagnose COPD and its severity (expressed as a percent of predicted value post-bronchodilator):

- Mild COPD: FEV1 is above 80%
- Moderate COPD: FEV1 is 50-79%
- Severe COPD: FEV1 is 30-49%
- Very severe COPD: FEV1 below 30%

5.3.1 Spiromtery Limitations

Spirometry is adequate for benchmarking external airflow which provides useful insight into the airway functionality and lung capacity. However, even if the patient has expired fully,

there is always some air left in the lungs, regardless of the exhale force. This is the Residual Volume (RV) and is usually about 20-25% of the FVC. The total Lung Capacity (TLC) is expressed in Equation 5.1:

$$TLC = FVC + RV \tag{5.1}$$

Unfortunately, it is impossible to measure the RV with a spirometer as it is not physically exhaled from the body. For this, a less convenient gas dilution or body plethysmography test must be performed. Furthermore, spirometry doesn't sufficiently evaluate intrinsic lung properties such as diffusion efficiency or capillary performance. In these scenarios, more sophisticated methods such as those shown in Table 5.3 must be employed. Most of these tests are associated with low risk and designed to be done in a standard physician's office, however, they are far less convenient and more expensive compared to spirometry.

Tab. 5.3: Pulmonary Function Tests

Test	Description
Spirometry	Measures the rate of air flow and vital capacity. Requires multiple breaths, with regular and maximal effort.
Body plethysmography	Measures the total amount of air the lungs can hold using an airtight booth called a plethysmograph with accurate flow and volume logging.
Lung diffusion	Assesses how well oxygen gets into the blood from the inhaled air via diffusion. Requires several minutes of causal periodic breathing. Typically, the diffusing capacity for carbon monoxide (DLCO) is measured.
Gas dilution	A person breathes from a container containing a known amount of a gas. The test measures how the concentration of the gases in the container changes.
Pulse oximetry	Estimates blood oxygen levels. Requires placement of a probe on a finger or another skin surface such such as an ear.
Arterial blood gas	Directly measure the levels of gases, such as oxygen and carbon dioxide in blood. These tests are usually performed in a hospital. Typically, blood is typically taken from the femoral artery via needle.

5.4 Quality Control

To reiterate, the main appeal of spirometry is the noninvasive, quick and easy nature of the procedure. However, there is a consequence to this simplicity. The correct technique, posture, and mindset of the patient are all important variables to the success of the procedure and the slightest mistake can invalidate a spirometry maneuver. Coaching helps standardize the process, but simply explaining the procedure does not guarantee compliance or repeatability. As a result, well defined, patient independent, quality control methods have become a standard in spirometry. These include post-trial error screening and session-based reproducibility rules.

5.4.1 Errors

Recall, that a correct spirometry testing procedure is characterized by [61]:

- Beginning the maneuver with maximal blast effort
- Applying maximum effort throughout the maneuver
- Keeping a tight seal on the spirometer mouthpiece to avoid leaks
- Avoiding variability in output flow, such as breaks
- Completing the maneuver in one continuous breath

Not only are these rules hard for a patient to remember and consistently implement, they are also very difficult for a clinician to enforce subjectively. Fortunately, many of the most common errors in the above rules can be identified via FV or VT curve data and reproducibility rules, provided the clinician is adequately trained. The following errors in Table 5.4 are the most common and critical that can occur in spirometry testing. For further reference, the graphic illustrations provided by the ATS and displayed in Figure 5.5, visually show examples of common errors, how they manifest in spirometry curves, and how to prevent them.

5.4.2 Reproducibility

In addition to checking for trial errors, it is also imperative to verify the independent trials in the session have enough similarity to be considered clinically significant. Furthermore, since only one set of metrics gets recorded for per session, criteria must be established for selecting or computing the most representative trial in the session. For a session to be considered reproducible, the ATS suggests that at least three trials demonstrate reproducibility with each other. If this is true, the "best" effort is recorded, where "best" is the effort that achieves the largest FEV1. If a session is found to fail the reproducibility requirements, it is either

Tab. 5.4: Common Spirometry Errors

Error	Description
Cough	Occurs when a patient coughs into the flow tube, the aberration caused is very prominent and easily detectable in the FV curve. Usually identified by a second, superfluous, peak inflow.
Slow Start	Occurs when the patient does not start with the maximum blast effort, resulting in a gradual rather than steep slope to the PEF. It is defined as occurring when the PEF occurs at a volume larger than 0.7 L, [cite David P. Johns et al. "National survey of spirometer]
Early Termination	Occurs when a patient ceases the airflow before the entire breath is exhaled. This appears as a discontinuous drop in flow to zero. It should be suspected if the FET is less than six seconds.

repeated, assuming the patient has the stamina, or it is put off for another time. The ATS asserts two spirometry tests are reproducible if [61]:

- The difference in value for the respective FEV1 values is less than 5% or less than 200mL
- The difference in value for the respective FVC values is less than 5% or less than 200mL
- The PEF variation is less than 10%

5.5 Conclusion

This chapter provided a specific overview of how clinicians utilize spirometers and their results to diagnose obstructive and restrictive respiratory diseases. While many of these guidelines may only be relevant to a physician or care provider, it is important to acknowledge the inherent procedural, rule-based nature of these guidelines, which are well poised for automation. From diagnosis to quality control, much of spirometry can be expressed in an algorithmic form. For these rules to be effective it is vital that the source signal (airflow) is measured in a precise, robust manner. Later chapters in this work will explore alternative modalities for measuring the coveted airflow signal using nothing other than a smartphone. The coupling of a novel measurement technique with the algorithmic decision rules outlined in this chapter has the potential to lead to an influential mHealth spirometry application that will significantly enhance spirometry making it available and accessible to everyone.

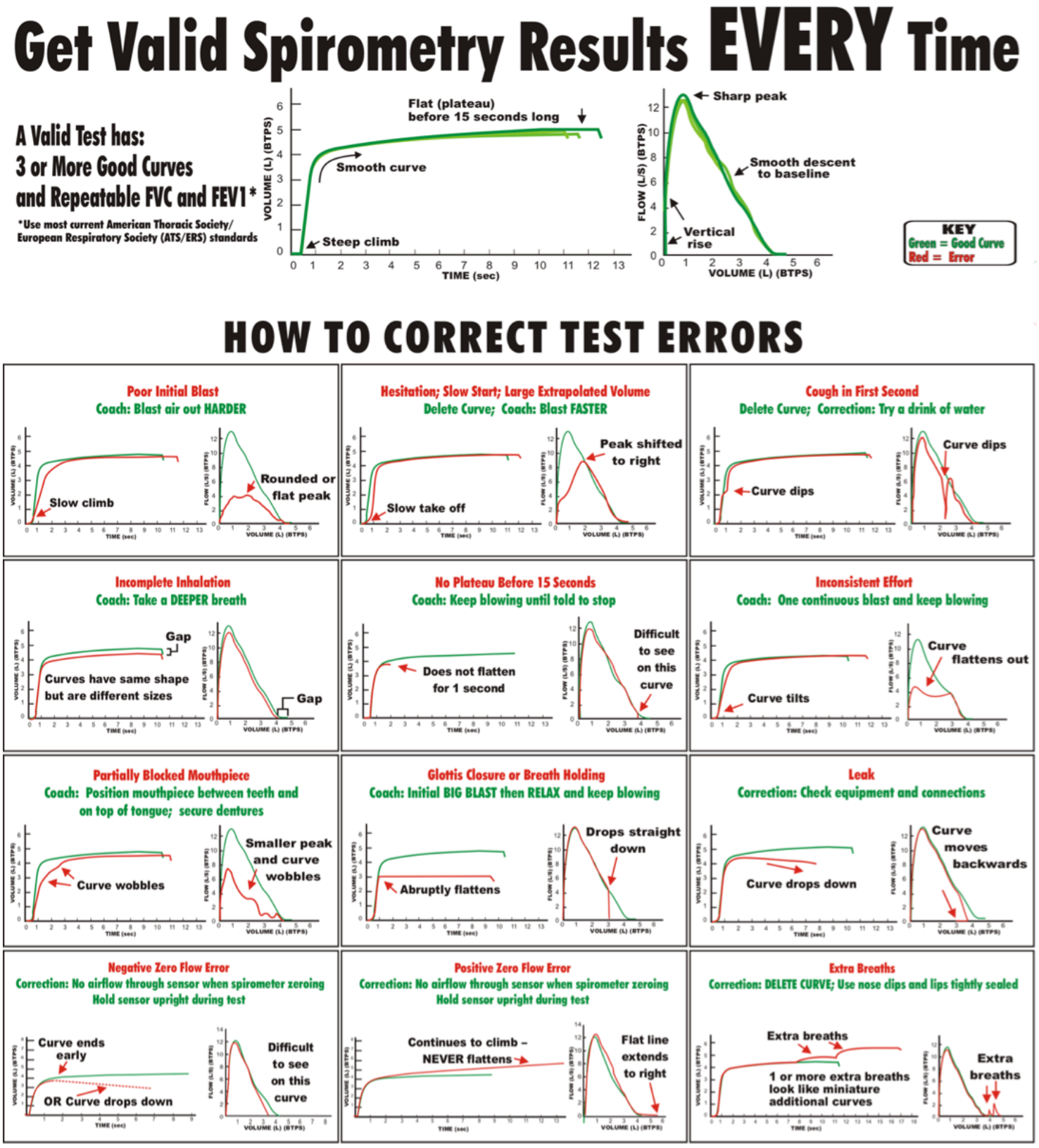


Fig. 5.5: Handout for spirometry reproducibility and error guidelines

5.6 Afterword

This chapter closes the captivating or to some, slumberous overview of the respiratory system and spirometry function testing. These preliminary chapters aim to serve as background and motivation for the main, technical contribution of this work which will be the topic of the remaining chapters. While it may have been gratuitous to start the discussion with events that occurred over 300 million years ago, the aim is to intuitively cover the functionality of the respiratory system so the anatomy, physiology and associated disease states are clear to a general audience. It is difficult to appreciate, let alone comprehend the utility of spirometry without a solid foundation of normal respiratory function. As mentioned, impactful solutions will only arise through multidisciplinary collaboration and a shared narrative, but first, the problem must be fully understood, and for a spirometry solution to be impactful it must be accurate, convenient and affordable.

6 Sound Background

> *Sound is the vocabulary of nature.*
>
> **— Pierre Schaeffer**
> French electronic sound pioneer

The majority of this work relies on capturing and processing sound in a way that maximizes extractable information. While a plethora of microphone technologies and processing techniques exist, the focus of this section will be limited to what can feasibly be done on a modern smartphone with no additional hardware. The chapter will begin by outlining the electro-mechanical process of converting sound pressure waves into a digital signal, then unfold into a discussion on how specific transformations can be applied to digitized sound to obtain additional context via feature extraction.

6.1 Microphones

Microphones, a type of sound transducer, have been around for over a century and convert acoustical energy in the form of sound pressure waves into electrical energy in the form of an audio sequence or signal. Many types and arrangements of microphones exist for different purposes. Up until the 1990's microphones were typically wired and handheld, but with the insurgence of digital hearing aids and mobile phones, a competitive market developed around the design of tiny low powered microphones. The current standard for these micro-sized microphones is known as a MEMS microphone and by 2010 they were utilized in the most popular smartphones, solidifying their seat in the world of sensors. In fact, according to IHS Inc, more than four billion MEMS microphones will ship in 2016 and will reach almost six billion units annually by 2019 [3].

6.1.1 MEMS Microphone

In 1986, the Defense Advanced Research Projects Agency (DARPA) published a proposal which first introduced the term "microelectromechanical systems", or MEMS for short [55]. The name, MEMS, is somewhat intuitive as they represent a class of electronics that integrate

mechanical and electrical components and have feature sizes ranging from micrometers to millimeters. Their small size and scalable manufacturability make it possible to integrate them into a wide range of systems, and smartphone manufactures do exactly that. Nearly all sensors and display components in modern phones involve some sort of MEMS technology.

In the case of smartphone microphones, MEMS technology is preferred for many reasons; they are smaller, cheaper, less power-hungry, and easier to fabricate and integrate into semiconductor packages. Furthermore, they have a high signal to noise ratio (SNR), resulting in cleaner audio capture at greater distances. Many advances in MEMS microphones have been tailored for smartphone usage as they are by far the biggest market. For example, they have a low power mode feature which allows them to always be listening for "Ok Google" without consuming significant battery power. In addition, modern Bluetooth headsets, laptops, smart-home assistants, cars and hearing aids also leverage MEMS microphones. Some technologies employ arrays of them and use a processing technique called beamforming to spatially locate the source of the sound.

However, MEMS components do have limitations. Because of the speech dominated use, MEMS microphones are optimized for speech clarity rather than reproducing original sounds as our ears hear them. As a result, they have poor sensitivity at low frequencies below 100Hz due to physical ventilation constraints and are hypersensitive between about 4-6kHz due to the Helmholtz resonance (the effect that makes empty bottles whistle in the wind). This is why MEMS microphone manufactures typically only specify the frequency response between 100Hz and 10kHz rather than the human range of 20Hz to 20kHz.

MEMS Design

The acoustic transducer fundamentals of a MEMS microphone are nothing new. They are basically a DC-biased capacitor, where movement of a membrane used by audio pressure changes the voltage over a capacitor plate or plates. This change in voltage represents the audio signal which is then digitized via neighboring Application-Specific Integrated Circuit (ASIC). Leveraging changing capacitance to measure sound pressure is certainly not a new technique as it was first shown in the invention of the condenser microphone by E.C. Wente of Western Electric in 1916 [39].

The design of a MEMS microphone relies on constructing a variable silicon capacitor and is shown in Figure A of 6.1. The capacitor consists of two silicon plates. One plate is fixed (the green plate) while the other one is a movable membrane (grey). External sound enters though the perforated holes of the solid fixed plate, striking the membrane and modulating the air gap comprised between the two conductive plates, thus changing the capacitance between the plates. A ventilation hole, allows the air compressed in the back chamber to flow out and consequently allows the membrane to move back. Aside from allowing the

membrane to vibrate, the internal chamber is also designed with specific acoustic resonance properties which characterize the frequency response and SNR [85]. In a typical smartphone, the MEMS microphone is housed in an intricate, acoustically engineered cavity between the printed circuit board (PCB) substrate and the exterior phone casing, usually along the bottom edge. To protect the membrane from dust and liquids, the ASIC, armored with an epoxy like "glob top", is placed beneath the pinhole inlets with the microphone placed adjacent as shown in Figure B of 6.1.

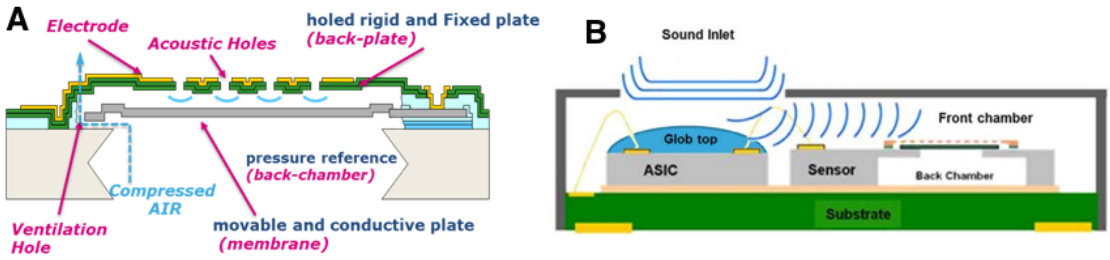


Fig. 6.1: A typical MEMS microphone assembly where (A) is the microphone itself and (B) is a common design pattern utilizing a MEMS microphone in a smartphone style enclosure.

The design of MEMS microphone and other membrane-based microphones is clearly borrowed from the superior engineered design of the human ear. In the ear, the eardrum (tympanic membrane) serves as the membrane and its movement is similarly transduced into an electromechanical signal by the cochlea. Rather than being processed by an ASIC, the brain receives the sound signal via nerve impulses.

Conclusion

At its core, a MEMS sensor is a variable capacitor that measures capacitance change between a rigid fixed plate and a movable membrane plate caused by the incoming wave of the sound. These tiny, sub-millimeter microphones achieve sound quality adequate for speech recognition and conversation but often lack in other domains. Their saving grace is low power whilst high SNR operation and cheap, scalable manufacturing process. Additionally, they are easy to integrate with other components to form powerful sensing packages such as those present in modern smartphones.

6.2 Digital Sound Processing

Once the audio signal is in the digital domain, its true potential can be unleashed.

The world is filled with signals: images from remote space probes, sonar echoes, seismic vibrations, voltages generated by the brain, and countless other applications. Digital Signal Processing (DSP) is the science of using computers to understand these signals. Some of

the common applications of DSP include speech recognition, image enhancement, data compression, filtering, neural networks, and much more. DSP is one of the most powerful technologies to date and continues to prove its power in new applications. This section will cover the process of going from analog to digital and the common DSP techniques applied to digital sound. These techniques are crucial in enabling the audio based machine learning solutions presented in this work.

6.2.1 Analog to Digital Conversion

In order to apply DSP to a problem, one must first obtain a digital signal. In the case of audio, this is either done with digital synthesis, i.e., generating a digital sound on a computer, or through digitizing an analog audio signal from perhaps a microphone. The process of digitization is typically done with an analog to digital converter (ADC) and will be the subject of this section.

Most signals in nature are continuous along a particular dimension: light intensity that changes with distance; voltage that varies over time; a chemical reaction rate that depends on temperature, etc. ADCs are devices engineered to allow digital computers to interact with these everyday signals. Digital information differs from its continuous counterpart in three respects: it is sampled, quantized, and often encoded. These steps are outlined in the subsections below and illustrated in Figure 6.2. The material in this section is summarized from the excellent DSP Guide website [78]. While sampling and quantization are fairly general, the encoding step varies in different applications. In the majority of current digital audio systems (computers, compact discs, digital telephony etc.), multi-bit Pulse Code Modulation (PCM) is used to represent sound signal in the digital world as it permits filtering, mixing and other potential manipulations to be easily applied.

Sampling

Sampling is the reduction of a continuous (analog) signal to a discrete (digital) signal. It is also called digitization of time. Sampling results in a sequence of samples, which are discrete in time but still continuous in amplitude. The sampling rate or fs is defined as the number of samples taken per second. A higher sampling rate results in more data points in the sequence and hence a higher resolution signal, but it requires more storage and is slower to process. Conversely, a lower sample rate results in fewer samples to describe the same analog signal and therefore results in a lower quality digital representation. The sampling rate places a fundamental limit on the type of information that can be accurately represented in the digital realm. Namely, the Nyquist theorem says that sampling rate should be double the frequency of the highest frequency signal. Any frequencies outside of the Nyquist limit are incorrectly represented in the digital signal as a form of distortion known as aliasing. It is

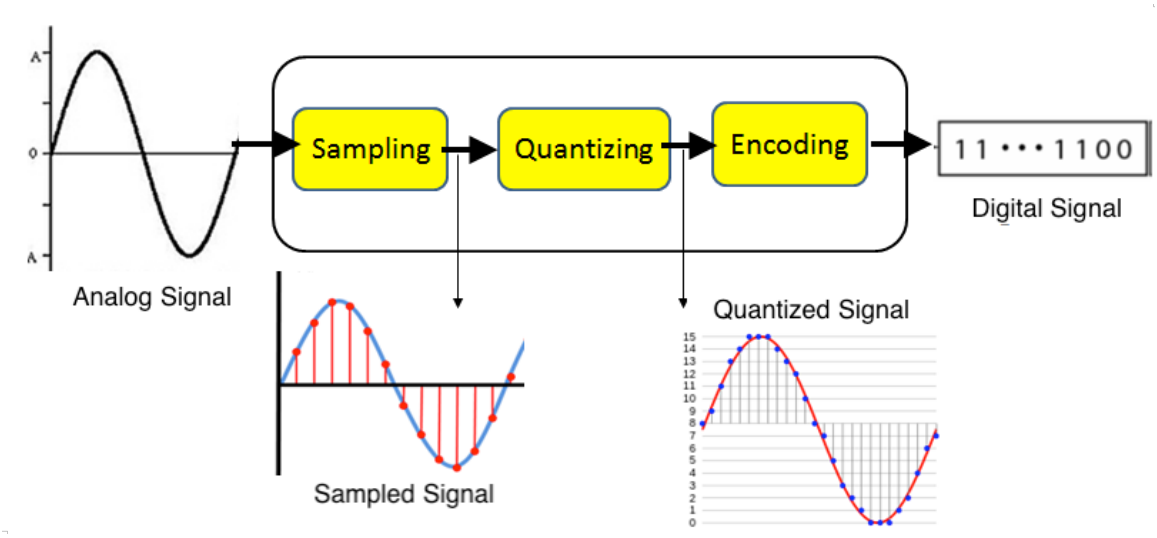


Fig. 6.2: Standard ADC pipeline from analog to digital

standard practice for ADCs to employ an anti-aliasing filter to prevent aliasing by removing frequencies beyond the Nyquist limit before sampling.

Adult humans can hear frequencies in the range of 20Hz to 20kHz, some newborns can hear up to 22kHz and dogs can hear up to 45kHz. Thus, in order to preserve the quality of sound sensed by the human ear, the Nyquist theorem dictates that a sampling rate of roughly 40kHz required, which explains why CDs and mp3 digital music files use a sampling rate of 44.1kHz.

In summary, fs is the only parameter that matters in sampling and the Nyquist limit places a bound on the minimum fs that can be used in order to prevent irreversible aliasing distortion.

Quantization

Quantization is the process of mapping a large set of input values to a smaller, countable set by rounding values to a fixed level precision. After quantization, the signal is discrete in both time and amplitude. The process that performs quantization is called a quantizer and the round-off error introduced by quantization is referred to as quantization error. The number of available discrete amplitude levels determines quantization error which depends on the number of bits used to represent each sample. If more bits are used to quantize a signal, its quality is improved, similar to using a higher sampling rate. For instance, an 8-bit sample will have $2^8 = 256$ discrete levels. In terms of SNR in the digtal realm, each additional bit increases the SNR by 6dB as represented in Equation 6.1, where N is the number of bits used to represent a sample. Common PCM samples are of 8, 16, 20 and 24 bits wide:

$$SNR\,(dB) = 6.02N + 1.76 \tag{6.1}$$

If the amplitude of a sample extends beyond the max limit that can be expressed by the quantization scale, it is clipped to the max value. A similar clipping effect can occur in the analog domain if a signal has more power than its amplifier can supply. In the majority of cases, clipping is unrecoverable and therefore a considered a form of distortion. An example of clipping is shown in Figure 6.3. To conclude, the main quantization parameter, N bits per

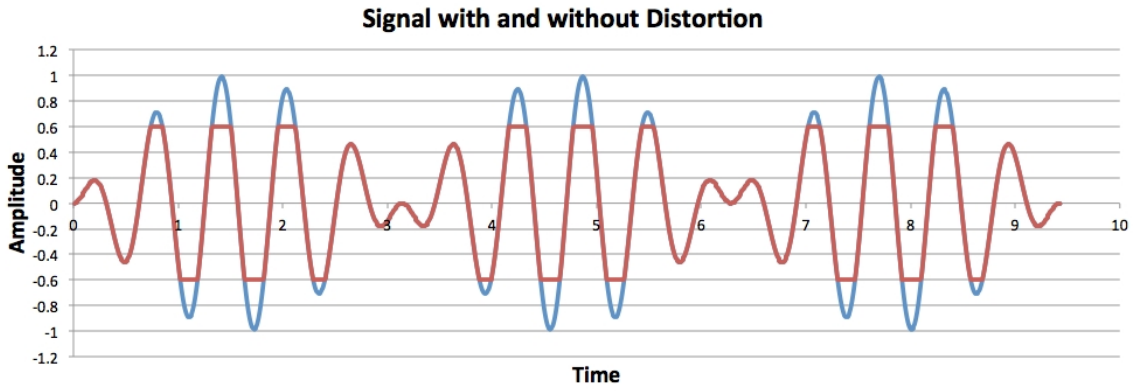


Fig. 6.3: An example showing the distortion due to clipping sinewave

sample, affects the SNR of the digitized signal and therefore affects the quality. Given an analog signal, the sampling process slices the signal along the time axis, and quantization dices it along the amplitude axis.

Encoding

The encoding process translates the sampled, quantized signal into 1s and 0s and embedded header information which gives the computer architecture instructions, such as f_s , that allow it to read the data. Usually, the encoding is defined by the computing architecture so it generally doesn't need to be considered.

Once an analog signal has been encoded into a digital format at the cost of unrecoverable quality loss, it is now ready for DSP. So far the focus has been representing an analog signal in a digital domain, but now the focus will shift towards various techniques that can be applied to a digital sound signal in order to extract useful information.

Conclusion

This chapter offers a high-level background of MEMs microphones and digital signal processing with a focus on sound. Next, time, frequency and time-frequency representations of sound are covered, as well as some of the features and insights that can be gathered from different representations.

6.2.2 Time Domain Processing

Traditionally, sound is defined as a variation in pressure waves and density caused by the propagation of the waves through a medium. Sound waves, being variation in air pressure over time, may be represented as a varying voltage or a stream of data over time. This is a *time domain* representation of sound. The amplitude represents the molecular displacement caused by the changes in air pressure which oscillate resulting in positive and negative fluctuations. In the digital domain, the amplitude is typically represented as a value between 1 and -1 which represent maximum positive and negative amplitudes of the signal, and 0 represents zero amplitude. An example of a spirometry exhalation using this characterization is displayed in Figure A of 6.4.

The time domain contains useful information despite its simplicity. For example, it is clear from Figure A of 6.4 that exhalation begins around $t = 0.5$ seconds and decays to near silence by $t = 3.5$ seconds. There are other ways to represent time domain waveforms. Figure B of 6.4 shows the amplitude envelope, which summarizes the change in amplitude over time in a more concise form. Amplitude can be directly interpreted from the waveform, but there is not enough information to adequately discern the type of sound creating the loudness. Without knowing the context of Figure 6.4, it could very well be a car horn or firework or any explosive sound with a decay. It is also common to represent amplitude with the decibel (dB) scale which is logarithmic and closer to the way the human ear perceives amplitude.

Examples of time domain processing include adjusting the amplitude, trimming to a specific time range, reversing, and changing the speed (which changes the pitch too). Basically anything that can be done on a basic cassette tape. The main descriptive features that can be easily extracted from the time domain are based on amplitude peak counting, timing such as overall duration, or duration above a particular amplitude. In most cases the amplitude envelope is sufficient for basic feature extraction. The envelope can also be smoothed, downsampled or fit to polynomial coefficients if the desire is to represent the envelope with a smaller set of numbers. As shown in the following sections, useful frequency based features can be extracted by converting from this domain into the frequency domain.

6.2.3 Frequency Domain Processing

The *frequency domain* refers to the sound representation with respect to frequency rather than time. Instead of illustrating amplitude versus time, a frequency domain representation shows how much of the signal lies within each given frequency band over a range of frequencies. The frequency information can be computed digitally using the discrete Fourier transform

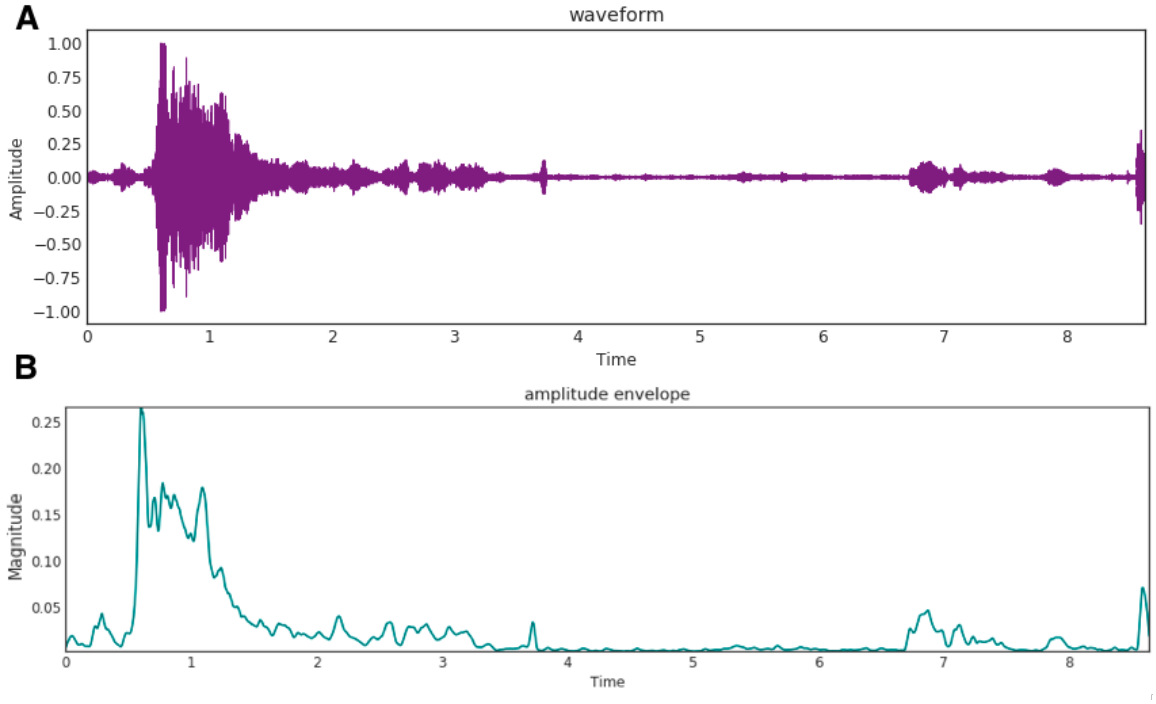


Fig. 6.4: A) An untrimmed sound recording of spirometry exhale, time in seconds. B) The amplitude envelope of the sound

(DFT) of the time domain signal and transformed back to the time domain with the inverse DFT. An example of a frequency envelope, known as a power spectral density (PSD) plot, is displayed in Figure 6.5. It is clearly a much different representation of sound and reveals that much of the exhale sound is low frequency below 1 kHz. There is a second peak around 4kHz which may describe a wheezing sound present during exhalation. Many sounds can be classified purely via frequency content. For example speech from an adult male, female and child can be distinguished by this representation.

The most common frequency processing usually involves converting to the frequency domain, filtering the frequency content, and converting the transformed signal back to the time domain. Frequency-based filtering can be done this way. Frequency domain audio features usually summarize the frequency information by grouping it into large frequency buckets, such as low, medium and high, and then associating a magnitude for each one.

6.2.4 Time-Frequency Processing

Both time and frequency domains involve a magnitude value in the y axis that varies along a single dimension of either time or frequency. It is possible to combine the information in both of these domains into what is referred to as a time-frequency (TF) representation of the sound. Other names and variations of this format include short time Fourier transform

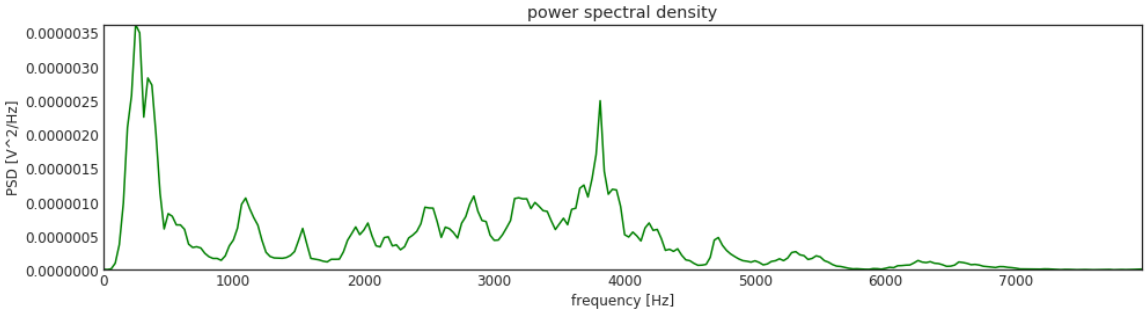


Fig. 6.5: A Power Spectral Density (PSD) plot of a spirometry exhale, which conveys the magnitude of frequencies within a specific range

(STFT), spectrogram and sonogram. A common format is a graph with two geometric dimensions: time and frequency; and a third dimension indicating the amplitude of a particular frequency at a particular time. This third dimension is typically represented by the intensity or color of each point in the image.

There are many variations of format which often depend on the discipline. For this work, time will be the x axis and frequency, the y. The frequency and magnitude axes can be either linear or logarithmic, depending on the primary purpose of the graph. The frequency axis has even more complex non-linear representations such as the Mel-frequency cepstrum scale, which scales the frequency axis similar to how human ears perceive; sounds between 100 and 4kHz are stretched out and other less audible bands are compressed. The third intensity dimension can also be scaled and represented in dB or other magnitude scales. Another form of extreme scaling is binary thresholding where any intensities below a limit are rendered white and all above black. This helps pick regions of interest from surrounding noise. Similar thresholding can be done to expose certain pitches or harmonics in the frequency axis. Examples of linear, Mel and threshold scaling are shown in Figure 6.6.

Most of the processing and feature extraction performed in the time or frequency domain can also be done in the TF domain, although some, such as amplitude peak counting may not be as obvious or are much easier to do in a simpler domain. Both trimming and filtering can be applied to the sound by cropping the x and y axis, respectively. Since frequency and time are both represented, this domain can be utilized for pitch tracking as well. Furthermore, there are mathematical transforms to go back to a time or frequency only domain, although some information may be lost.

The TF domain is ideal for classifying different types of sounds as most sound types have a time-varying component such as amplitude decay in addition to a pitch varying component. Figure 6.7 shows how different qualities of sounds manifest in the spectrogram. Notice how some sounds are classified from the time-varying component, such as the gunshot, while others are obvious from the frequency content such as the dog bark. Finally, some sounds,

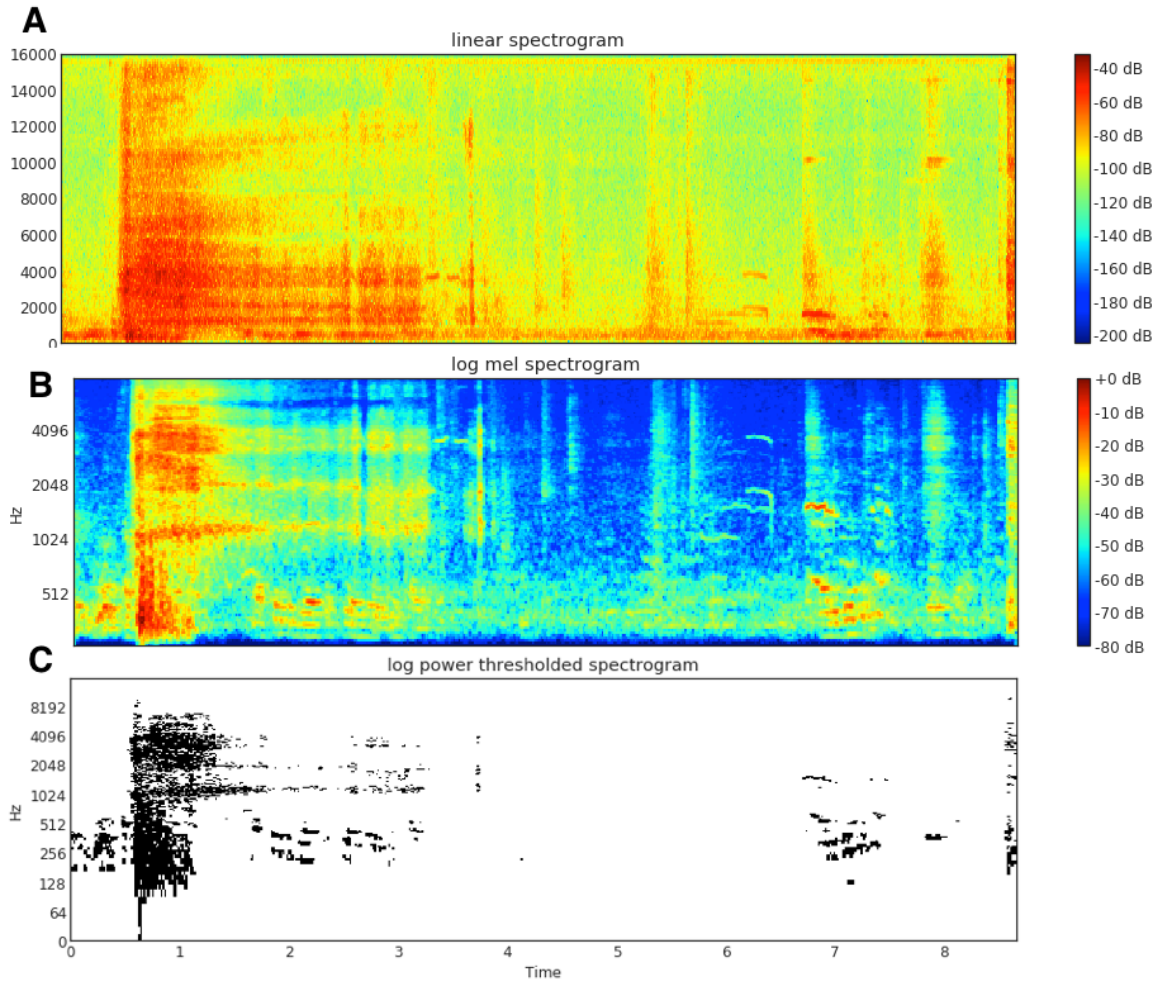


Fig. 6.6: Various time-frequency spectrograms with different scaling. In A) the frequency axis is scaled linearly B) employs log-Mel frequency scaling and C) applies a magnitude threshold to a log-scaled frequency axis

such as the siren require both the time and frequency variations to easily recognize. This work relies heavily on spectrograms and the Mel-scaling is the representation of choice for both classifying a spirometry maneuver as legitimate or not, as well as regressing to the flow versus time curve given the TF representation. This work will later show the spectrogram, along with other manually extracted features from the sound, inherently have a massive amount of predicted power when when used in the context of spirometry, especially if machine learning is leveraged to wrangle all of the features in an optimal fashion.

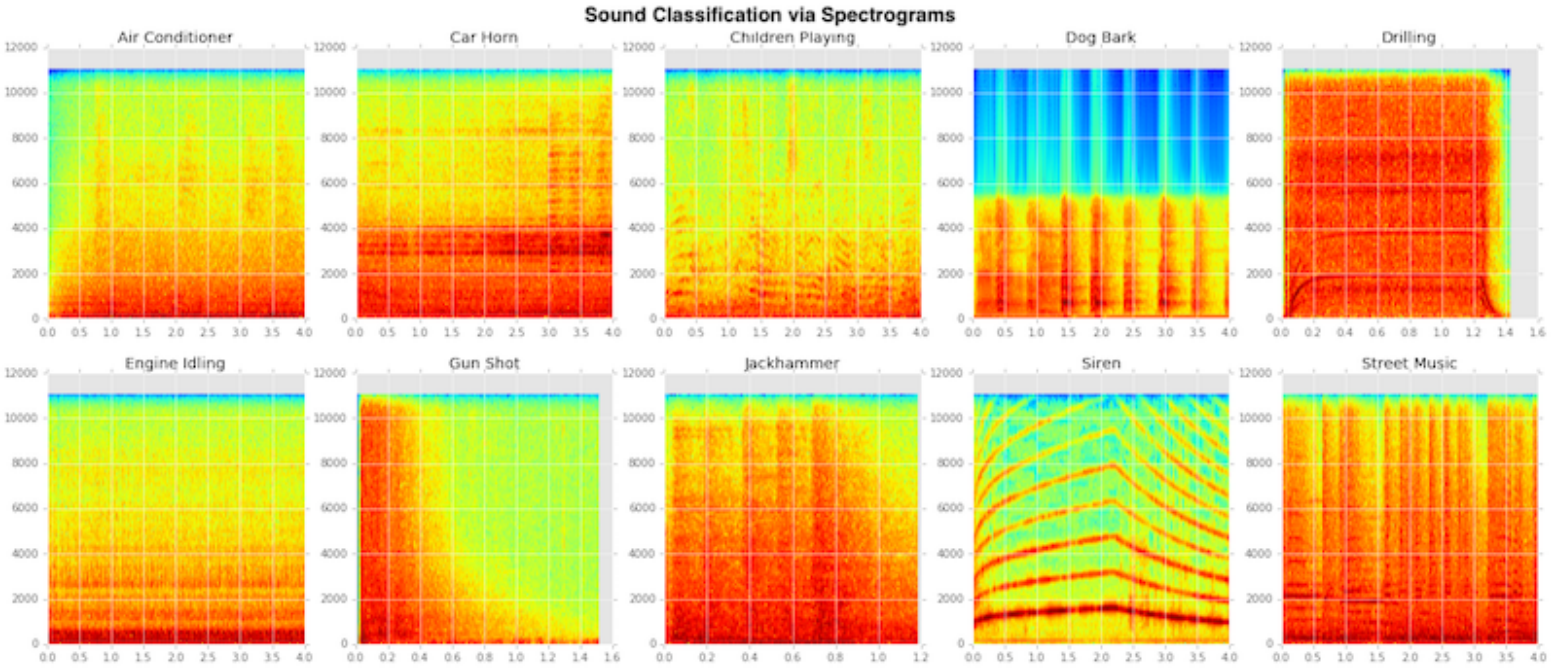


Fig. 6.7: Different sound classes from the Urban sound dataset plotted as a linear spectrogram, where time (seconds) is the x axis and frequency (Hz), the y axis.

7 Airflow Physics

Since the crux of this thesis relies on the ability to sense airflow via sound, this section will cover prior work in this area from a general sense, and also within the context of spirometry.

Understanding the relationship between sound and airflow is a common challenge in many fields. Perhaps the first researchers that come to mind given the topic of airflow sensing are atmospheric scientists. They are faced with the difficult task of understanding weather patterns which includes the need to measure extreme hurricane winds in an accurate manner. Interestingly, there are many other disciplines interested in understanding the effects of wind, from high-end tent manufacturers to military projectile firms. Thanks to the multidisciplinary longing to understand airflow, there is a massive amount of prior research on the topic. While much if it is far too application specific to extrapolate to human-powered airflow, some of it is quite generalizable to a model that pairs well with the physical design and functionality of a smartphone MEMS microphone. Before going into the various physical models that fit the scope of this problem, it is important to understand the limits of human physiology.

7.1 Physical Constraints

There is a speed limit that restricts the maximum velocity of air particles exhaled by a human. While there is little research directed at this theoretical question, empirical evidence gathered with high-speed motion camera suggests this limit is roughly 5 m/s in free air, which drops off beyond about 3 ft from the source [81]. This research was focused on coughs and sneezes, which is certainly on the extreme side of airflow output. Forced exhale is somewhere between an impulsive cough and a constant flow, so while the peak velocity isn't expected to rise, the propagation distance for a forced exhale should be greater since far more volume is being moved at a similar speed and thus will travel further before dispersing.

Additional insight can be obtained analyzing the PEF of thousands of patients or using the predicted value given patient parameters such as height. PEF is measured as a flow rate

rather than velocity, Equation 7.1 shows the conversion between flow rate and velocity when the radius of the pipe (airway-opening) is known.

$$flow\ rate = \pi r^2 v \tag{7.1}$$

Since we are interested in the maximum limit, a peak flow of 11.2 L/s will be used which is 10% higher than the maximum predicted PEF value for a 6 foot 5-inch 35-year-old male. This also aligns with the findings in the upcoming Dataset chapter in which the maximum recorded PEF of 40 thousand trials is 11.32 L/s. In the case of spirometry, the pipe radius is designed to be the radius of the mouth in order to capture the flow from the mouth "as is". The guidelines for spirometer testing suggest a breathing technique similar to fogging up a mirror via breathing, so it is fair to assume a pipe radius of approximately 0.75 inches plus or minus a 0.25 inches. Using $r = 0.75$ and *flow rate* = 11.2, the max velocity, is found to be $v = 9.82$ m/s, versus the 5 m/s limit found in free air. Note, this number is highly dependent on the mouth radius which is difficult to control when the spirometry tube is omitted. This insight reveals that in order to approximate spirometry flow via airflow in free air, some sort of mathematical transform must be developed to convert free airflow into the traditional pipe airflow seen in spirometry. This transform will be depended on the mouth radius r and the mouth to phone distance x. In other words, a transform needs to map airflow at the microphone membrane to the flow measured at the mouth. The application in this work assumes the smartphone microphone will be pointed down, normal to the airflow and approximately an arm's length away. This distance x will arbitrarily be chosen to be 20 inches for the purposes of the modeling math.

In summary, it can be assumed that air velocity near the microphone source will range from 0 to a maximum less than 10 m/s and more in the ballpark of 5 m/s. The microphone will be held approximately 20 inches (0.5 m) away from the subject's mouth which as an approximate radius of 0.75 inches (0.02 m). Before diving into the transform required to map airflow at the mouth and flow at the microphone, it is important to first demonstrate it is indeed possible to measure airflow in the range of 0 to 10 m/s range using a MEMS microphone.

7.2 Physical Models

This section will broadly survey various approaches to promising airflow modeling that may be applicable based on the constraints outlined in the last section.

Prior work in acoustics has done an excellent job modeling the effect of external wind on a microphone membrane with the primary purpose of creating optimal windscreens. In order

to design a windscreen with optimal noise attenuation characteristic, the dynamics of wind and the rigid sphere design of the microphone exterior must be investigated. While the goal of designing a wind attenuator is in opposition to the objective of this research, the theory gives rise to the relationship between airflow and recorded sound.

Tab. 7.1: Fluid Dynamics Terms

Term	Definition
Bernoulli's principle	states that an increase in the speed of a fluid occurs simultaneously with a decrease in pressure or a decrease in the fluid's potential energy in order to adhere to the law of conservation of energy.
laminar flow	a smooth flow where each particle follows an uninterrupted path, never interfere with one another. Occurs at low Reynolds numbers, where viscous forces are dominant. Characterized by smooth, constant fluid motion.
turbulent flow	an irregular flow characterized by tiny whirlpool regions. Occurs at high Reynolds numbers and is dominated by inertial forces, which tend to produce non constant velocities such as chaotic eddies, vortices and other flow instabilities.
bluff body	a body that has separated flow over the majority of its surface as a result of its shape. In other words, a body which when kept in fluid flow, the fluid does not touch the whole boundary of the object, but instead leaves a wake which causes drag. A school bus is a bluff body compared to an aerodynamic Lamborghini.
Reynolds number	a dimensionless value measuring the ratio of inertial forces to viscous forces used to describe the degree of laminar or turbulent flow. Larger Reynolds number results in more turbulent flow. Correlated with velocity.
stagnation point	a point in a flow field where the local velocity is zero and the total, stagnant pressure is maximized.

7.2.1 Airflow Microphone Model

A sphere, like that of a traditional microphone head, can be modeled as a bluff body because flow with enough velocity hitting one side of the sphere will not touch the opposite side and instead leaves a wake. This is true with a smartphone microphone as well. As the flow speed increases, Reynolds number also increases which leads to a more turbulent wake. For this described model, it is assumed the flow is incompressible and the directional effects of wind hitting bluff microphone sphere are ignored (one-dimensional flow). The time-dependent

stagnation pressure term is conveniently given by the equation defining Bernoulli's Principle and is independent of the sphere's radius:

$$P(t) = \frac{1}{2}\rho V(t)^2 \tag{7.2}$$

This can be expanded using Reynold's decomposition to be expressed in terms of the average laminar flow velocity, U, and the fluctuating turbulent velocity magnitude $u(t)$:

$$P(t) = \frac{1}{2}\rho U^2 + \rho U u(t) + \frac{1}{2}\rho u(t)^2 \tag{7.3}$$

So what can be learned from this? Since $P(t)$ is the maximum pressure on the body due to the deflection of airflow, then fluctuating air velocity can, therefore, give rise to fluctuating stagnation. Furthermore, when the spherical object contains an embedded pressure sensitive membrane, similar to a microphone, this stagnation pressure has a dominating effect on the sound picked up by the sensor. Therefore, the theory suggests wind speed can be tracked with a microphone by exploring the stagnation pressure which governs the magnitude of the pressure experienced by the membrane.

In other words, it is expected that sound amplitude recorded via the microphone is proportional to wind speed when constrained to an ideal, rigid spherical model with one-dimensional flow. This is empirically verified by anyone who has tried to record audio when it is windy. In order to actually understand the transformation from recorded sound to wind speed, especially on a spectral level, experiments must be conducted as the effect is completely dependent on the bluffness and geometry of the microphone sensor and enclosure. The next section gives an example of such an experiment.

Infrasonic Wind Speed Measurement

Developing accurate methods to measure the speed of airflow, specifically wind, at the source of the microphone has been a research topic for about as long as portable microphones have existed.

Past research by NASA and the government dating back to the 1960's used a simple infrasonic (sound below the limit of human hearing, i.e., < 20Hz) technique, as well as the theory above, to measure the speed of the wind. After recording wind of known speed with a special microphone limited to the 1 to 20 Hz range, researchers were able to accurately measure wind speed from 2 to 7 m/s simply by measuring the overall loudness of the recording within the narrow 0 to 20Hz band [11]. Their findings showed a roughly 5 dB increase in loudness in the infrasonic bands for each 1 m/s increase. It is also intuitive to suggest faster wind would result in greater modulation in the microphone membrane and thus a louder recording. Other research has verified this and also shown the loudness to speed relationship begins to fail or becomes more complicated as the frequency range is increased into the

audible sound domain [58]. While the findings alone are not directly applicable since MEMS microphones can not measure sound fluctuations below 100Hz, the general concept can still be utilized.

In the case of airflow, the two main components that contribute to the noise recorded by a traditional microphone are the *natural* turbulence of the airflow and the *manufactured* turbulence generated when the natural air strikes the microphone assembly and disrupts the membrane. Several potential models of this are motivated by simulation and controlled experimentation as presented in "A Review of Wind-Noise Reduction Methodologies" [87]; however, these are not discussed here because they cannot be directly transferred to the modern MEMS microphone as they are very specific. They do show that with the right model one can learn a great deal about the airflow from the spectral density of the sound recording. Unfortunately, in most cases, the spectral region of interest is within the infrasonic range when the air velocity is between of 0 to 10 m/s. This conclusion is also elegantly present in nature. Wind can indirectly generate infrasound through its turbulent nature. This infrasound, or slow pressure variations, are what create ocean waves. As wind speed increases, so do the sizes of ocean swells, but not necessarily the frequency of the wave.

To conclude, the turbulence created from airflow gives rise to infrasonic pressure waves which manifest as low-frequency noise when picked up by a microphone. As airflow velocity and turbulence (Reynolds number) increase, the infrasonic waves and perceived noise is magnified. While there is not an obvious change in the frequency content as velocity increases, some complex models reveal a subtle but measurable effect, mostly in the infrasonic bands.

Audible Band Wind Speed Measurement

An experiment is conducted and documented in Section 10.1 which tests how well this simple loudness based method transfers to a modern smartphone. The findings suggest the method is feasible for measuring constant airflow flow in the 0 to 4 m/s range with a MEMS microphone. The experiment also showed this method could only be used to track steady-state airflow and is not suitable for rapid variable flow.

Ultrasonic Wind Speed Measurement

Another experiment, also outlined in Section 10.1, attempts to measure airflow using ultrasonic sound outside of the human audible limit near the upper limit of what MEMS microphones can detect (between 20kHz and 23kHz). In this experiment, the Doppler shift due to the airflow is investigated as a potential way to measure the airflow. This experiment is still in process so results have not yet been reported. It seems a large barrier to this approach is that physical movements such as phone wiggling or body motion can also add a lot of noise in the ultrasonic region of interest where the airflow is also occurring. Other work has successfully used ultrasonic Doppler shifts to measure airflow in air conditioning ducts, which doesn't have the issue of external motion other than air [71]. Ultrasonic

processing can also be used to measure breathing rate for people sleeping; however, this technique measures chest expansion rather than actual airflow [4]. With stereo microphones it may possible to measure airflow by analyzing the phase shift between the two microphone sources, assuming their displacement is known. This is also part of the active ultrasonic experimentation being conducted.

Conclusion

The goal of this section is to investigate if airflow can be tracked from a microphone and the answer is yes, followed by a shrug since the relation is very application specific. Next, a spirometry specific transformation will be explored that will map airflow at the mouth to the flow measured at the position of the microphone.

7.2.2 Airflow Mouth Dispersion Model

The above model sheds light on how airflow velocity at the microphone source can be sensed. The issue of converting flow measured at the microphone to flow at the source of the mouth still remains. Prior work involving SpiroSmart, approximated this transform as a spherical baffle in an infinite plane where arm distance and head circumference must be known [46]. The aim is to model the head as a sphere in an infinite plane with air flowing around it to approximate an exhale. The inverse of this model allows one to compute the pressure at the exterior of the sphere (where the mouth would be) given the pressure an arm's length away. The main limitation of this model is it assumes airflow out of a pipe (mouth) is the same as airflowing around a sphere.

Another approach, which will be adopted in this work, models airflow from the mouth as flow through a pipe which disperses into an infinite plane or open channel similar to the depiction in Figure (b) of7.1. This is known as a turbulent jet model and is a standard particle dispersion statistical model which assumes the turbulent flow yields particle dispersion in a plume-like manner when averaged over time. The laminar flow is streamlined and loses velocity as the frictional components generate turbulence. The extent to which this occurs depends on the Reynolds number. At any given time, a snapshot of the flow will appear random due to the turbulence, as shown in panel (a) of Figure 7.1, but the true distribution becomes clear when integrated over time. This type of mixing model is used for many general particle dispersion such as smoke or heat and was first suggested by fluid dynamics pioneer GI Taylor, nearly a century ago [82].

As mentioned before in Equation 7.3, the total velocity is the sum of the laminar flow, U, and the turbulent flow, $u(t)$. In the case of an exhale, it is difficult to know the proportion of each component at any point in time, however, it can be certain that there is a component of both. Anyone who has observed a smoker exhaling can confirm the smoke stream forms a

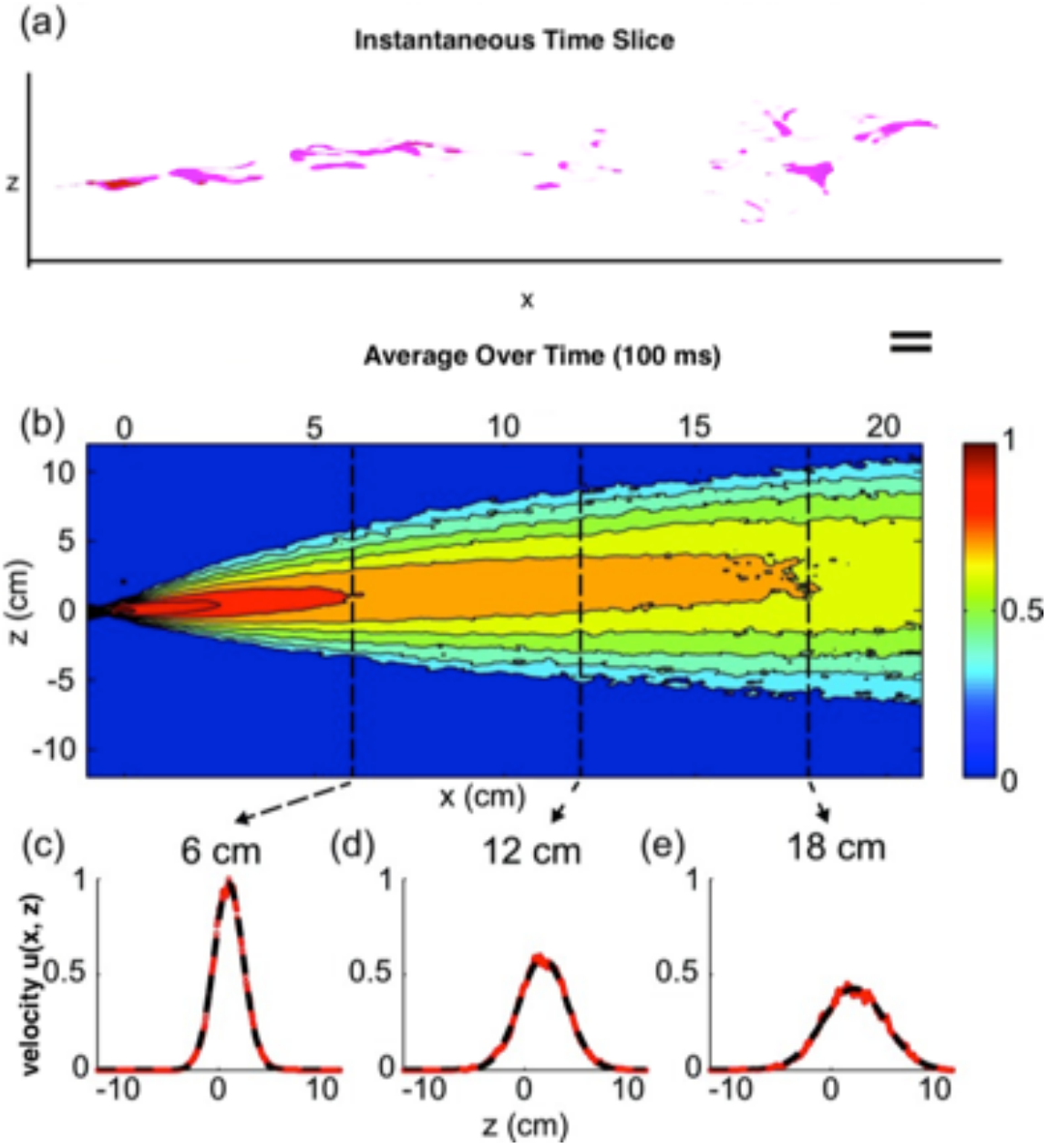


Fig. 7.1: (a) Typical experimental image of the instantaneous particulate distribution as illuminated by the laser sheet. Scale bar is 1 cm. (b) Empirical contour plot of the time integrated particulate intensity. Red denotes high particulate concentration, blue denotes zero concentration. (c)–(e) Cross sectional profiles of intensity vs. spatial displacement take the form of a Gaussian. Source: Turbulent dispersion via fan-generated flows

plume-like trajectory rather than a narrow streamlined one due to the conversion of laminar to turbulent flow, yet still has a general trajectory due to the source laminar component. In this model, the plume width increases with distance from the pipe source, while the particle velocity decreases with distance. Therefore the relationship between velocity and plume width at a given distance can be modeled as a Gaussian that traces the spatial intensity along

the axis normal to the flow of air and with a width, σ, proportional to plume width ($4\sigma \approx$ plume width). This is depicted at three different distance slices in Figure (c) of 7.1 [31].

The normally distributed velocity at a given distance slice, x, can be parameterized as a scaled Gaussian:

$$u(x, z) = u_{max} \exp\left(\frac{-z^2}{2\sigma^2}\right) \tag{7.4}$$

Where $u(x, z)$ is the velocity at a distance x from the pipe, z is the distance from the centerline and σ is the plume width. The σ at an arbitrary x is not known, but it can be approximated if the pipe orifice diameter, d is known and the open air channel medium is made up of a quiescent particle of similar properties of the exhaled particles. In this case, the plume adapts a conical shape and laboratory observations reveal that all turbulent round jets possess the same opening angle, regardless of fluid, orifice diameter, and initial velocity [16]. The universal value is 11.8 °, yielding a cone radius distance ratio of: $tan(11.8) = \frac{1}{5}$. Note, $x = 0$ must be set at the incident angle a distance $\frac{5d}{2}$ into the pipe (Figure 7.2). Since the cone radius at any x is half the width of the plume width, which is $\approx 2\sigma$, it can be said that $\sigma \approx \frac{x}{10}$. This outcome can be substituted in for σ into Equation 7.4. Now the velocity at any point in the plume is expressed as a function of u_{max}.

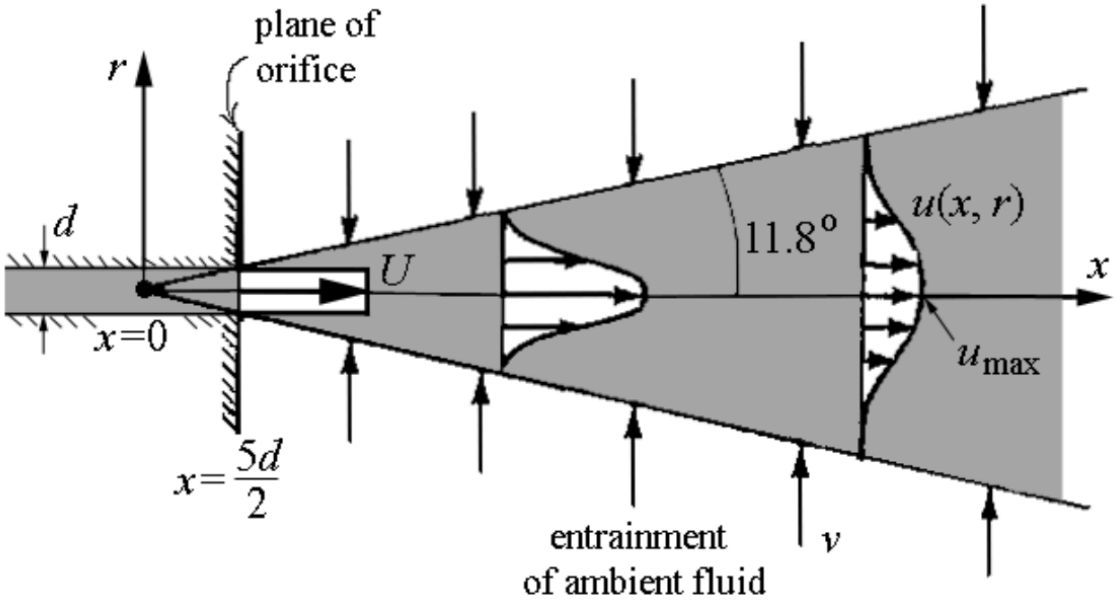


Fig. 7.2: Typical jet turbulence model with the universal angle shown. Note this figure depicts what was defined earlier as z as r [16]

By acknowledging there are no external acceleration forces and the only source of momentum is from the jet stream exiting the pipe, u_{max} can be derived by integrating along the cone volume such that momentum is conserved. The result is shown in Equation 7.5. Also note that average velocity is $\frac{u_{max}}{2}$.

$$u_{max} = U\frac{5d}{x} \tag{7.5}$$

In other words, the velocity along the centerline of the jet indeed decreases inversely with distance from the pipe source. The empirical result collected from studying several fans suggests a slightly different result. It was found that the plume takes more of a paraboloid shape rather than a simple cone. This is due to extra turbulence induced by the fan blades. If this were also the case in an exhale, $u_{max\ fan}$ would decrease according to the inverse square of x as follows:

$$u_{max\ fan} = U\frac{ld}{\sqrt{x}} \tag{7.6}$$

Where l is an unknown mixing length constant that describes the contribution of rotational turbulence from fan blades or in our case the respiratory system. Since the shape of the plume expansion and the turbulence contributed by the respiratory system are unknown without proper experimentation, it is impossible to know the true velocity decays as a function of x. The goal is to understand the physical limits, so the ideal, linear decay will be used in the next section as it would results in a larger velocity at the microphone than the inverse square decay. Either way, with the ideal model, the velocity at any region within the wake of the exhale plume can be computed assuming the pipe or mouth diameter is known and the velocity at the lips is known. This model can then be inverted to provide the velocity at the lips given the velocity measured near the phone microphone.

Applying the Physics Model

The following constants are taken from the Physical Constraints section above and applied to Equations 7.5 and 7.6:

$$U_{mouth} = 9.82\,m/s$$
$$d_{mouth} = 0.04\,m$$
$$x_{mouth\ to\ phone} = 0.5\,m$$
$$x = 0.5 + \frac{5d}{2} = 0.6\,m$$

$$u_{max} = 9.82\frac{(5)(0.04)}{0.6} = 3.2\,m/s$$
$$u_{plume\ avg} = \frac{3.2}{2} = 1.6\,m/s$$

So based on the ideal Airflow Mouth Dispersion Model, the max velocity at the microphone source during a peak flow instance for a large male can be expected to be 3.2 m/s assuming the microphone is aligned with the centerline normal to the mouth surface. If the phone's position is offset from the centerline, the velocity can be approximated using the scaled Gaussian distribution.

Conclusion

In both the microphone and dispersion model, several assumptions were made in order to shed light on a physical model that satisfies the constraints of the problem. It would be naive to assume these models will hold up in reality, but at least they give a sense for what to expect. Furthermore, in order to unify the models, several more factors must be considered. For example, the effect of the turbulence created near the microphone due to the blunt wall surface of the phone, or the effect of the hand and arm holding the phone. Also, a form of non-linear amplification of the microphone signal by the phone operating system could further complicate the modeling. Nonetheless, the exercise is a useful step to begin understanding the difficult fluid dynamics that characterize the airflow interaction that occurs between the mouth and the microphone membrane.

Given the apparent complexity in the task of modeling airflow from the mouth to how it is sensed on a MEMs microphone and presented digitally, it is reasonable to turn to machine learning to use data to fine-tune and fit the optimal model to this elaborate problem.

8 Machine Learning Background

> *With great power comes great responsibility.*
>
> **— Uncle Ben**
> The Amazing Spider-Man comic

8.1 Introduction

Machine learning is a subfield of artificial intelligence (AI) focused on making AI learn from their experience. Contrary to typical control theory or rule-based algorithm development, in machine learning (ML), instructions are not provided for the machine to follow. Instead, examples of the intended behavior are collected and the machine ingests the information using a learning algorithm, then assembles what it learned into a program or model that attempts to mimic the intended behavior. Ideally, the computer learns to be clever rather than relying on a clever programmer to explicitly code up its behavior. The key word here is "ideally". Often times humans must intervene and spoon feed the exact, relevant information in the form of manual features. The key to any machine learning problem, or learning in general, is the data. In this work, there is a complete chapter dedicated to the dataset used in the work (see the Dataset chapter).

Machine learning has become the standard tool for solving many complex problems from speech recognition to photo object detection. It is often compared to a human brain as both the brain and an ML algorithm are very good at detecting specific patterns that are understood after copious amounts of data have been ingested. There are problem areas where machine learning is not ideal or preferred, such as when enough data is hard to obtain, or a mathematical solution already exists. For example, it is very difficult for ML algorithms to learn the concept of a Fourier transform (described in the Sound Background chapter), even though the algorithm was first published over 200 years ago. The key to machine learning is knowing when to use it.

Regarding sound based spirometry, as shown in the Airflow Physics chapter, modeling how airflow is sensed by the MEMS microphone is extremely complex with many constants and geometries either unknown or tedious to measure. On the other hand, recording the sound

of airflow in an experimental environment is simple. When data is easier to gather compared to a physical model, machine learning is a promising solution. This is even more so when the data contains complex information as in the case of sound. Machine learning is also generalizable to a wide variety of environments and has been shown to perform well in domains not well represented in the data it learned from. For example, an English speech recognition AI can be retrained on a relatively tiny corpus of French vocabulary and still outperform any human tuned algorithm [38].

8.1.1 Types of Machine Learning

At its core, machine learning is a methodology of statistical guessing and it involves many algorithms and problem types. This section will provide a brief overview of the main types of machine learning. This work is mostly based on supervised learning, although some critical tasks are solved with semi-supervised and unsupervised learning.

Supervised Learning

A supervised learning system involves a "teacher" that provides example inputs and their desired outputs. The goal is to learn a mapping from inputs to outputs that also works well on new inputs (generalization). This is machine learning with an asterisk as it requires constant human intervention and labeling. Supervised learning is commonly used as it typically works well for many problems and is amazingly effective when sufficient (>10,000) labeled training examples are available.

Semi-supervised Learning

The goal of semi-supervised learning is to discover characteristics of a dataset when only a subset of the data is classified or labeled. Perhaps 10% of a 100k dataset has labels, then an ML model can be trained on the labeled portion then attempt to classify the unlabeled portion. Following this, human labelers can correct some of the modeling errors on the unlabeled set (which probably does much better than random guessing), then feed it back through. With each iteration of the model comes improvement, and it may eventually reach a point where it is trusted to achieve its goal, well before the 100k set is manually labeled.

Unsupervised Learning

Unsupervised learning draws inferences from datasets consisting of data without labeled responses. This technique can be used to discover patterns in data without upfront intervention or guidance. Clustering is one form of unsupervised learning. A clustering algorithm may naturally learn the difference between a sunrise photo and a selfie simply from the arrangement of colors or edges. Or in the case of shopping recommenders, clustering is used to group shoppers based on purchase history, then suggest products based on mutual shopping patterns within the cluster.

Reinforcement Learning

A reinforcement learning algorithm (RL), or agent, learns by interacting with its environment in a free form. The agent receives a reward when it correctly completes a task and is penalized if the task is failed. An agent typically learns without human intervention by maximizing its reward and minimizing its penalty. This is the type of learning used in AI game players such as chess bots or an automatic video game players. In these cases, the reward is points, or winning, or surviving. The key is for a human to correctly identify this reward. A bad example of a reward would be to instruct an autonomous driven agent to simply, "never crash". It would very quickly learn that the optimal strategy is to not move.

8.1.2 Common Machine Learning Tasks

Another way to think about machine learning is by the tasks, or use cases, that it needs to perform. Table 8.1 summarizes such tasks within the spirometry context. This work mostly utilizes binary classification for spirometry effort detection and regression to transform the audio signal into a flow curve and metrics.

8.1.3 Input Features

Recall from the Sound Background chapter there exist many ways of representing the sound that is innately conducive to different insights and feature extraction. There are two ways to build a set of extracted features for an ML model: manually or automatically. The manual method is more traditional as it allows engineers to bake their expertise into the problem by extracting intelligent relevant features. The automatic method is data-driven, rather than engineer driven. It is usually implemented with a deep neural network (DNN), more specifically a convolution NN (CNN). A raw form of the input is supplied in a DNN, such as a spectrogram image, and the DNN learns which parts are meaningful or worth analyzing in an automated way. This distinction is expressed in Figure 8.1.

8.2 Classical Machine Learning

In the manual case, engineers preprocess the input data, sound in this case, and convert the high dimensional sound data (thousands of discrete samples) into a more manageable table of features. Such features may include, number of peaks, duration, low-frequency loudness, and average loudness. Each feature is a single value such that the set of features for all sound can be expressed as a table where columns are features and rows are different sound files. This feature matrix, X, has N unique sound rows and M features and must be paired with a ground truth table, Y which has N rows, but a single column of the desired output.

Tab. 8.1: Common machine learning tasks in the context of spirometry

Data Driven Question	Machine learning task of choice
Is this sound a spirometry exhale? How certain is the prediction?	**Binary Classification**: Classify the elements into two groups on the basis of a classification rule. Often times the predictor answers a true/false question as being a true if there is a $>$ 50% likelihood, otherwise it is determined as false. In this case 50% is the decision threshold.
Does this sound contain speech, coughs, airflow, or something else?	**Multi-class Classification**: Classify instances into one of multiple classes. Similar to binary classification, typically a likelihood is assigned to each class and the class with the highest likelihood is selected.
How healthy are my lungs?	**Regression**: Predict a continuous value for each example, such as a score between 0 and 100.
What is the airflow vs time sequence given this sound sequence?	**Sequence to Sequence Regression**: Predict a set of continuous values given a separate sequence. Language translation is another example.
Given several unlabeled exhale sounds, which ones have wheezing?	**Clustering**: Organize data into common subsets or clusters in an unsupervised manner.
Are these exhale sounds from the same person?	**Factorization**: Matrix factorization techniques and other supervised or unsupervised embedding models can be used to provide a global similarity metric between any two items. Also used for person detection.

With this notation, the ideal ML algorithm (with unlimited data), f, perfectly maps input X to output Y such that: $Y = f(X)$. The machine learning process aims to create the ideal f by trying an arbitrary f' to obtain a prediction for Y, defined as Y'. In order for f to do a good job approximating Y, two rules need to be established.

First an *error function*, E (also called a cost or loss function), must be defined to evaluate how well the approximation Y' matches the ground truth Y. The goal of the ML model is to formulate f in order to minimize $E(Y, Y')$. In regression, E is usually mean squared error (MSE), and in binary classification, E is typically cross-entropy (also called log loss). There are several cost functions to choose from, but for this work, only these two are used unless otherwise stated.

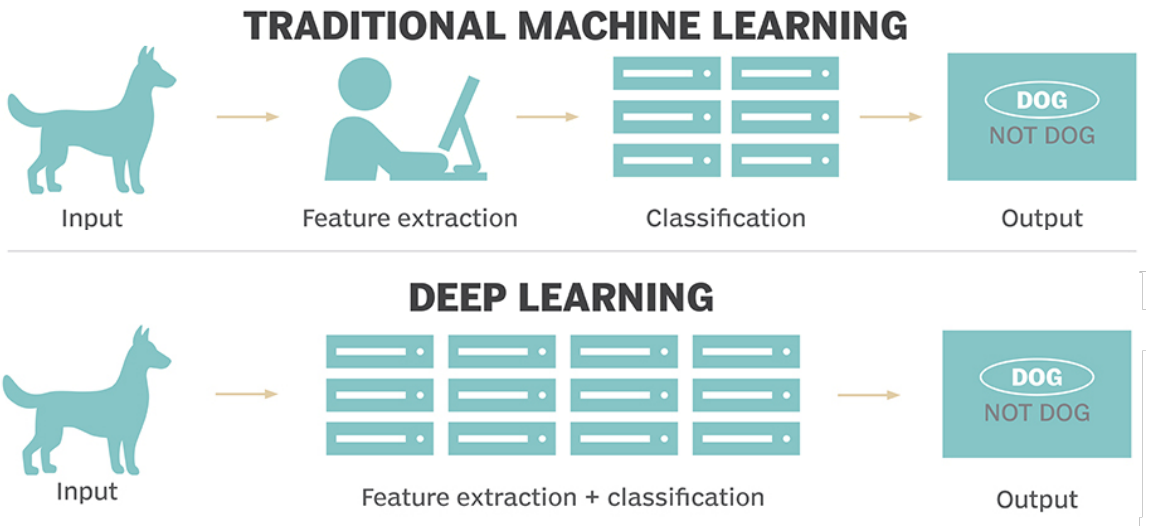


Fig. 8.1: Visual indicating the difference between manual and automatic feature extraction

Second, an *update rule* must be defined. Minimizing the cost function is an iterative, optimization process. The ML algorithm suggests a f and then E can be computed. The update rule must define how f should be tweaked for the next iteration based on the E of the current iteration. The goal of the update rule is to generate a new f that achieves a lower E than the prior version of f. Unlike the cost function which is dependent on the problem being solved, the update rule is usually defined by the ML algorithm being used, a common one being stochastic gradient descent (SGD). It turns out there are a few different basic types of ML algorithms for both classification and regression that inherently have different update rules and subsequently different strategies for developing a strong model, f. Before going into the different manual machine learning models, a simple regression example will be described to solidify what was just stated.

Applied Example

Joe is an aspiring, but hot-tempered golfer who has placed his trust into his patient robot coach, TigerBot. Joe's goal is to master hole 5 by landing his shot on the green, rather than in the trees, sand or water. Every time Joe swings, TigerBot receives swing features, X. To keep it simple, X has two columns, x_0 quantifies swing speed and x_1 quantifies the swing angle. TigerBot does not have a way of seeing where the ball lands, but fortunately, Joe very clearly expresses his feelings through speech making it possible for TigerBot to know how bad the shot was based on the amount of cursing and grumbling perceived. Therefore, Y' is expressed as the anger quotient where a large value means Joe is very upset and the desired Y is the minimal anger quotient of 0. Using squared error as a cost function, the cost function for TigerBot to minimize can be defined as $E = (0 - Y')^2$. Following a shot, TigerBot gives Joe advice via update function, such as "swing twice as hard", or "adjust the angle a bit to the left". Despite what Joe may think, to TigerBot he is just a function, f that produces an output Y' from a shot with features X. Eventually, after perhaps 100 attempts,

Joe will be ecstatic as he has finally landed the ball on the green. Alas, E is now much smaller and Joe gets to go home a happy, improved golfer while TigerBot can rest assured the job it was designed to do has been done.

In this example, the key concepts for a standard machine learning algorithm are applied. Furthermore, it shows that an ML algorithm doesn't need a highly accurate ground truth metric to measure and minimize error. An ideal, physics model might have information about the exact trajectory of the ball and use that to tell Joe exactly how to swing. In contrast, ML uses weaker, but easier to obtain information, such as the sounds of disappointment, to reach a formidable solution provided enough training examples are provided. The next section will dive into classical ML model types used in this research.

8.2.1 Linear Models

Linear models are the simplest to understand and are best when the relationship between X and Y is linear.

Linear Regression

In the regression case, a linear model attempts to fit a straight hyperplane to the dataset (i.e., a straight line of best fit when X describes a single feature). Typically, a one-dimensional linear model fits a bias b and a weight vector w to the data to satisfy: $y' = wx + b$. In multiple dimensions this may be expressed in vector notation as shown in Equation 8.1:

$$Y' = W^T X = b + \sum_i w_i x_i \tag{8.1}$$

Where the bias b is treated as the first value on W and a placeholder 1 is inserted as the first value of X for simpler notation. To prevent overfitting, regularization techniques like LASSO (L1) and Ridge (L2) are often used. Regularization will penalize input features that are less useful, by forcing their corresponding weight to a near 0 or 0 value. Usually, the penalization factor can be specified as desired. It is almost always worth including regularization because it permits post-training analysis which can identify the useful features (with higher valued weights) and those that can be ignored. Typically the weights are optimized using the gradient descent algorithm (SGD), which is briefly covered in Artificial Neural Networks.

Logistic Regression

In the classification case, Logistic Regression (yes, the name seems contradictory) is used to draw a linear hyperplane to separate classes rather than fit a trend as in linear regression. When a new sample is presented, the logistic regression algorithm looks at where the sample falls relative to the decision plane and estimates the probability of the sample falling into a certain class (this is the regression part). The probability is computed using a softmax

function, which is also common in neural networks and described later. Given the probability, p, a decision threshold, T will output a 1 if $p > T$ otherwise, a 0. Usually 0.5 is used as T. When the decision is more complex than what can be represented by a linear plane, other approaches must be used. One such approach, support vector machines (SVM), has the ability to fit curved and other non-linear distributions, as shown in Figure 8.2. Other non-linear approaches include decision trees and neural networks.

8.2.2 Decision Trees

A Decision Tree is a supervised predictive model that can learn to predict discrete or continuous outputs based on the values of the input features it receives from a set of simple questions. It is similar to the Twenty Questions guessing game where a guesser might ask questions like "Is it alive?" or "Is it furry?" to continually narrow the solution space down with each question. Part of the strategy in Twenty Questions is to correctly order the questions: the first few questions should be broad so as to eliminate a large number of possibilities, while the last few should be more specific to hone in on the "best" possible answer. A decision tree works the same way where the inquiries are related to the input features.

The decision tree algorithm has an natural tree representation. The tree begins with a root followed a series of branches whose intersections are called nodes and terminal ends are called leaves, each corresponding to one of the classes to predict. The depth of the tree defines the maximum number of nodes before terminating at a leaf. Each node of the tree represents a rule specific to an input feature, that can be phrased as a question. During training, a decision tree tunes the questions and the order at which they are asked in such a way that each node corresponds to the rule that best divides the set of initial features. Given the natural hierarchy of nodes, it is possible to extract the importance of each input feature based on the ordering of questions. This is useful for feature selection in a similar manner to using regularization in linear models. It is also very easy to interpret how the model makes a prediction as the tree can simply be plotted and traversed. Finally, decision trees have the ability to model non-linear distributions as shown in Figure 8.2. Overfitting is a common problem with these models as they can be too curious and develop overly specific rules that do not generalize well to new data. They are therefore somewhat unstable compared to linear models, although some of the variants address this issue with clever tricks. The set of questions followed by their ordering is critical to the success of a decision tree model; however, it is nearly impossible to identify the perfect strategy. There are several tree based ML models that employ different techniques to achieve the ideal tree model. The two methods used in this work are Random Forests (RF) and Gradient Boosting Model (GBM). Both RF and GBM are ensemble methods, which means the model is a combination of several smaller decision tree models, but they have fundamental differences.

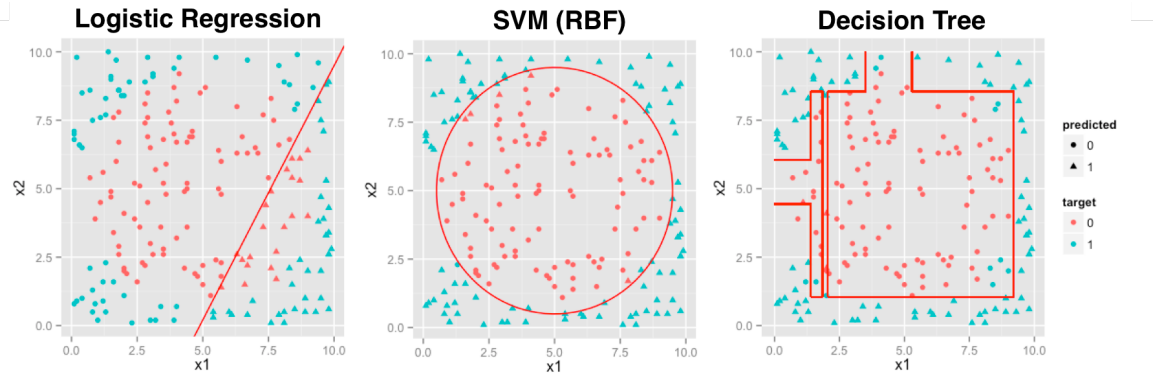


Fig. 8.2: Different classification algorithms applied to a non linear binary decision distribution, clearly the linear fit is not adequate and the decision tree is prone to overfitting, while SVM is well equipped to offer a generalized solution

Random Forests

The idea behind RF is to build many small trees that use a random subset of the features and then combine them into a "forest" of trees by employing a voting based update rule. Each tree by itself is a weak predictor, but combining many of them often yields a much stronger model. Since decision trees are highly prone to overfitting, each weak tree will overfit the data in a different way, and through voting, the differences are averaged out and the strength of the consensus increases.

Gradient Boosting

In contrast, GBM is more sophisticated. The idea is to again, combine weak predictors, but the trick is to find areas of misclassification and then "boost" the importance of those incorrectly predicted data points to prioritize fixing the error, then repeat. The update rule essentially uses gradient descent to decide which tree to boost and by what magnitude. While RF trains a new independent predictor each iteration creating a forest of trees, GBM only creates one tree which is iteratively improved. The resulting GBM model is often much smaller and faster than the RF equivalent but usually requires more data for adequate performance.

8.2.3 Clustering

As mentioned in Table 8.1, clustering is the task of dividing the dataset into a number of groups such that data in one group shares similarities that differ from the data in other groups. The groups are typically organized by sets of features highly correlated to a particular outcome. For example, athletes who have features such as long legs and thin body types may end up in a cluster that correlates to runners, while short legs and light body weight athletes end up in a jockey cluster. Clustering does not have to be supervised. In the athlete example, the same cluster could be formed without knowing the correct sport. Having labels, of

course, allows for the model to have more predictive power and certainty that it is learning the desired classification or regression relationship.

k-nearest neighbors

K-nearest neighbors (KNN) is one such algorithm that clusters all of the training data based on the features. A new prediction is computed by exhaustively comparing the new input features to every training data point and choosing the top k data points with the highest similarity. There are various ways of measuring similarity; however, the final prediction is based on the average prediction of the top k data points. This algorithm assumes two similar Y outputs also have highly similar X feature sets. When this is the case, KNN works very well but it is very vulnerable otherwise. Since all of the training data must be searched for every new prediction, this method does not scale well in scenarios involving large datasets.

K-means

An unsupervised form of clustering known as K-means clustering, attempts to sort unlabeled data into K separate sets. Each set contains a centroid, and the distance between the centroid and all the individual points in the set is minimized. For example, clustering a large set of documents based on the contained words may naturally sort them into categories that represent business reports, personal notes, and bills. K-means is used in this work to sort the large dataset by timezone in order to reveal the original clinic where each sample was collected (see Chapter 11). Aside from clustering data based on similarity, it is important to note the K in K-means algorithm is completely different than the k in KNN despite sharing the "k".

8.2.4 Sanity Check

It is a good idea to verify a trained model is actually doing a good job. In many cases, there may not be an existing benchmark for a machine learning problem so it is therefore wise to formulate a benchmark using a very simple, predictive model. If a more advanced model performs only marginally better this, it may suggest that the problem is either too difficult or the features are suboptimal. If one of these simple models performs surprisingly well, it may suggest an inherent correlation between X and Y that can be exploited without the need for machine learning.

In regression, a useful sanity check is to compute the mean Y for all inputs and then measure what the error would be if the mean were guessed on every sample. In the classification space, it is easy to compute the probability of guessing the correct class as $1/N_{classes}$. A better classification technique than random guessing is to use Naive Bayes, which builds a model based on the probability of each class and then guesses based on this distribution. For example, if the training data has 80 apples and 20 bananas, a Naive Bayes model would

make predictions assuming these statistics remain true in the future data and guess banana 20% of the time rather than 50% in the case of random guessing. If a trained ML model is not much better than one of these basic attempts, it is not doing a very good job learning and there is most likely an issue with the data or problem in general. This type of sanity checking has helped identify errors and flaws with the methodology used in this work that may have otherwise slipped through the cracks.

8.3 Artificial Neural Networks

This section will build a foundation for artificial neural networks as they are heavily explored as a potential solution in this work. The hope is to convey how artificial neural networks automatically extract powerful features by employing thousands of tunable artificial neurons inspired by how the human brain operates. Artificial neural nets are commonly grouped with artificial intelligence, so the discussion will start by outlining the recent commercialization efforts of artificial intelligence, as well as its theoretical foundations.

Artificial Intelligence

The concept of artificial intelligence is continually working its way into modern society. The news claims it will take everybody's jobs, businesses boast it is saving them millions, and tech lords peg it as the biggest threat to humanity. It was once a topic of philosophy and fantasy entertainment, but recently has now crept into many people's daily lives. Arguably the most useful services provided by search engines, digital content providers, email, navigation apps, shopping and personal assistant hubs are powered by AI. Despite its ubiquity in everyday life, it remains elusive.

When products are advertised as "battery powered" it is patently obvious to the average consumer how the product is differentiated from other, perhaps "wired" counterparts. When a product is branded as "AI-powered" there is no consensus as to what is entailed. When something is prefaced with "artificial" it is typically a manufactured copy of a physical, well-understood equivalent. But intelligence is neither physical or well understood and as long as this is true, artificial intelligence will remain a nebulous term with few implications. In fact, once a form of artificial intelligence is mainstream enough, it is no longer considered intelligent, rather it is simply another form of computing, simply reflecting on the fact that all luxuries are doomed to become necessities. For this work, intelligence will be pragmatically defined as *the ability to perceive information, and retain it as knowledge to be applied towards adaptive behaviors within an environment or context* [91].

Brain Analogy

Based on what has been observed to date, anything naturally intelligent possesses a brain. Furthermore, a brain with more connections, or a larger neural network, is generally considered more intelligent. Following this logic, researchers postulated, anything processing artificial intelligence must have some sort of artificial brain with similar functionality as a natural, human brain. With this came the concept followed by the crude implementation of an artificial neural network.

Much of how the human brain works is still a mystery, yet research is uncovering new insights at a rapid rate. For example, it is known that the most basic element of the human brain is a specific type of cell called neurons. Unlike the rest of the body, neurons do not appear to regenerate. Given this, it is assumed these cells are what grant us the ability to remember, think, recognize patterns and learn from experience. All 100 billion of these cells throughout the body can connect with hundreds of thousands of other neurons. A neuron itself is not much of a mystery anymore and can be classified in about 100 different ways, but the ways they work together and function as a brain is still unknown. What is known is the power of the brain comes from the complex connections between neurons, neuroplasticity which allows neurons to evolve and improve functionality over time, and most importantly, the sheer numbers of neurons which form the neural network.

A typical neuron receives information from other neurons through a host of thin receivers called *dendrites*. The neuron sends out spikes of electrical activity through a long, thin pipeline known as an *axon*, which splits into thousands of terminal branches, as shown in Figure A of 8.3. A neuron can be thought of as an information gate which produces an output along its axon i.e., it "fires" when the collective effect of its inputs reaches a certain threshold, called the *action potential*. The axon from one neuron can influence the dendrites of another neuron across electrically stimulated junctions called *synapses*, as illustrated in Figure C of 8.3. Some synapses will generate a positive effect on the adjacent dendrites which encourage its neuron to fire and propagate the action potential, while others will produce a negative or neutral effect. Learning occurs by adaptively changing the firing threshold which adjusts how one neuron affects there others.

A software-based artificial neuron is similar and depicted in Figure B of 8.3. It consists of a processing element which has a number of input connections, each with an associated weight. A transfer function operates on the weighted sum of the inputs to determine the output, which is then connected to adjacent artificial neurons. Therefore, an artificial neural network, Figure D of 8.3, is a network of interconnected, artificial neurons, although the scale is dwarfed compared to human neural networks. The artificial network is trained by iteratively adjusting the multiple weights of each neuron such that the network produces the correct output for a particular input in the training data.

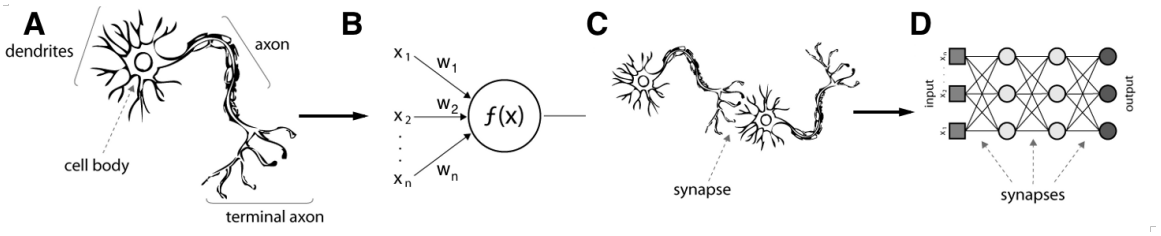


Fig. 8.3: Illustrates the artificial brain analogy. A neuron (A) compared to an artificial neuron (B) where both have multiple inputs, which influence the output through some embedded function, the artificial neuron has weights expressed as w and inputs as x. Neurons can be combined as in (C) to form a neural network and communicate via synapses. Similarly artificial neural networks (D) are a combination of artificial neurons.

The brain analogy is somewhat fragile and many scientists caution against taking it too seriously. Artificial neural networks try to model the low-level functionality of the brain; however, the goal is not to emulate a brain, but rather to resemble the properties of the brain that allow it to learn from experience. Whole brain emulation is a challenge reserved for computational neuroscience and there is still a lot of work that needs to be done before we have fully functional conscious brains in our pockets. Humans are not conscious of the low-level electrochemical processes going on underneath their skin, but the external effect, whether emotion, thought or action, is certainly apparent. The argument for the neural network approach to AI is that, if the low-level activities can correctly be modeled, the high-level functionality may be produced as an emergent property. This is in direct contrast with the traditional approach to AI which employs rule-based symbolic reasoning to model the high-level reasoning processes of the brain. From this point on, neurons and neural networks (neural nets or NN for short) will refer to the artificial variant, not the biological human brain.

History

Artificial neural networks are not a new concept. The earliest work goes back to the 1940's when McCulloch and Pitts introduced the first neural network computing theoretical model [57]. Also around this time Alan Turing laid out several criteria to assess whether a machine could be said to be intelligent, now known as the "Turing test" [84]. In the 1950's, Rosenblatt's work resulted in a two-layer network referred to as the perceptron, which was capable of learning certain classifications by adjusting connection weights [73]. Although the perceptron was successful in classifying certain patterns, it had a number of limitations. For example, Papert and Minksy were unable to solve the classic XOR (exclusive or) problem among other things [62]. Such limitations led to the decline of the field of neural networks for several years.

In the 1980's, researchers showed renewed interest in neural networks and began to create improved perceptron inspired networks consisting of several perceptrons arranged in multiple layers, now called artificial neurons. There were two key findings that stemmed from this

research. First, it that using an algorithm called backpropagation, one could efficiently update the weights of several neurons in multiple layers [74]. Second, it was theoretically demonstrated that multi-layered neural networks had the ability to learn any function. This was known as the universal approximation theorem. Still, these networks did not easily scale as they took weeks to train and usually proved no better than existing simpler. Neural networks came back into the spotlight in the mid-2000's and in order to differentiate them from previous iterations, researchers began calling them "deep" neural nets because they combined many more trainable layers than the less capable "shallow" nets of the past. Training deep neural nets (DNN) became known as deep learning.

The true breakthrough came in 2012 as a result of several key ideas and resources coming together. By this time graphics processing units (GPU) were incredibly powerful matrix calculators thanks to the sharp rise of computer gaming and later, Bitcoin mining. These GPUs also allowed neural networks to train much faster on complex data such as images or audio. Also around this time, a large database known as ImageNet containing millions of labeled images of pretty much everything was created in 2010 and published by Fei-Fei Li's group at Stanford [75]. This sparked yearly research competitions where researchers and companies such as Microsoft and Google battled to push the state of the art in large-scale image classification. In the first two years of the contest, the top models had error rates of 28% and 26%, respectively. However in 2012, Alex Krizhevsky et al. entered a neural network based submission, named AlexNet, which nearly halved the existing error rate to 16% [43]. Its success came from a combination of several novel ideas that would become crucial in further developing deep learning. These advances included parallel GPU training and rectified linear units (ReLU).

Since 2012, deep learning has gone mainstream and with it has come massive high-quality datasets on just about everything, programming frameworks like Tensorflow that make training and evaluating a deep learning model trivial, and much more powerful GPUs. Now every component that makes up a neural network has hundreds of variants optimized for different problems and data formats. The accelerating growth of GPU computing speed versus more traditional CPUs is shown in Figure A of 8.4. The ImageNet competition results are also shown in panel (B) of Figure8.4. Some interesting insights from this data reveal models with lower errors tend to have more layers (permitted by constantly improving GPUs), and since 2015 the DNN models have started to outperform humans at image labeling.

8.3.1 Artificial Deep Neural Networks

For a conceptual idea of how a deep neural network learns, imagine a factory line. After the raw materials (raw training data) are inputted, they are then passed down the conveyor belt, with each subsequent stop or layer extracting a different set of high-level features. If the

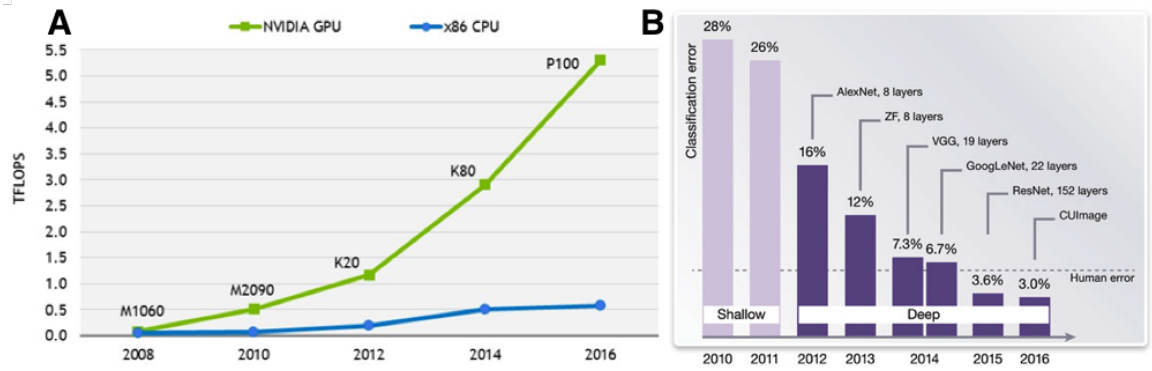


Fig. 8.4: Two ways of tracking the growth of deep learning. Figure A shows the growth of GPUs in terms of their floating point operations per second (FLOPS) compared to a CPU. Figure B shows the rapid improvements in the ImageNet including the recent surpassing of human level object detection

network is intended to recognize a dog breed from an image, the first layer might analyze the brightness and colors of its pixels. The next layer could then identify any edges in the image, based on lines of similar colored pixels. Following this, another layer may recognize textures and shapes as collections of edges, and so on. By the time the fourth or fifth layer is reached, the DNN will have created complex feature detectors where clusters of neurons can identify specific details such as floppy vs straight ears. The final layer before the output might be an embedding consisting of neurons that describe the input as a dog with curly white hair, a long snout, straight tail and floppy ears, i.e. a poodle.

These detectors, capable of extracting highly complex features, start as a blank slate. With the help of backpropagation, gradient descent (explained later) and thousands of labeled images, the weights of the neurons morph to strengthen and weaken certain connections such that by the end of training, neurons pass different types of information forward through the layers. Within a layer, neurons operate independently on the same set of inputs. The result of each is broadcasted to every neuron in the next layer. Because the tuned neurons dictate which information is passed along the network, it can be said that neural networks automatically extract high-level features, which is in direct contrast with the classical ML methods which require manual feature extraction.

Layers

There are three main types of NN layers: input, output and hidden:

The *input layer* must have a neuron per subdivision of the input., i.e., if the input is an image of size 100x100, the input layer must have a neuron for each pixel (10000). If the input is 1 second of sound with a sample rate of 16kHz, then there must be 16000 neurons at the input. For this reason, it is common to scale down images or downsample audio to reduce the size (number of neurons) of the input layer.

The *output layer* size must represent the number of desired outputs. If it is a dog detector only a single neuron is needed to specify true (1) or false (0). If it is a dog breed classifier, 100 outputs would be needed assuming there are 100 possible breeds to select from. In general, the problem becomes more difficult if the number of inputs or outputs is increased as there is more input information to process and more room for error in the output. Anything that can be represented as a matrix or vector can be considered as an input or output to a neural net. Images such as photographs or spectrograms, written text, and sensor time-series data such as sound or temperature are all common. Also, tabular data such as the manually extracted features described in Classical Machine Learning can be used as an input and the NN will likely learn a model that often outperforms the classical methods.

The *hidden layers* are all of the layers between the input and the output. They can be specified as any size, although there are rules of thumb to how hidden layers are arranged and sized, which will be discussed later. As mentioned, generally more layers yield a more powerful network but only when sufficient training data exists. With a marginal amount of training data, 3 layers may perform just as well as 10. There are different types of hidden layer architectures, which will be covered later in this chapter under the Architectures section.

An example neural network was previously illustrated Figure D of 8.3. This network consists of an input and output with 3 neurons and 2 hidden layers, also with 3 neurons. This example may be misleading, as the layers do not need to all be the same size. Usually, when referring to a NN, the number of layers counted omits the input and output. In this example, the NN has 2 layers of size 3. For the purposes of this work, the computation starts with the input layer and passes values to neurons in the feed forward (left to right) direction. More advanced networks such as Microsoft's ResNet apply skip connections which allow certain layers to ignore other layers [32]. It may look like the neurons send out multiple values because there are multiple lines extending from the neuron, but really there is still only one output value per neuron and it is simply copied or broadcasted along each of its output connections. Neurons always output one value, no matter how many subsequent neurons it sends its output to. Furthermore, neurons within the same layer are independent and do not process information sharing connections.

Loss Function and Update Rule

As with classical machine learning, neural networks have a loss (also called cost or error) function which measures the error of the model in its current state. The loss function is typically MSE for regression and binary cross-entropy for binary classification (same as classical machine learning). The goal of neural net training is to optimize the weights in a way that minimizes the loss function. The optimization strategy, or optimizer, is the driving force behind the update rule which must update all of the weights in the network with each training iteration. For all multi-layered neural networks, backpropagation is the update

strategy as it is capable of efficiently and exhaustively updating each weight in a tailored, systematic way. While there are several choices for optimizers in a NN, they are all more or less based on SGD. Usually mini-batch SGD is preferred over a full SGD because it requires less memory per iteration and computes the gradient faster as it bins the training data into small batches (usually 20 to 200 entries each) rather than computing the gradient for each training entry (which may range from 1000 to 10000 entries).

SGD has a few parameters. When mini-batches are used, the batch size must be specified. SGD also has a tunable learning rate (or step size) parameter, α, which dictates how aggressive the weight update will be. A low α means the network will train very slow, but thoroughly. If α is too large, the weights may never converge because they cannot be fine-tuned to the ideal configuration. This concept is easily understood in a two-dimensional optimization case. Imagine starting at an arbitrary point in a 2D bowl (Figure 8.5) and taking a step of size proportional to α, the goal being to stop when the slope of the bowl (the gradient) is 0. This point represents the optimal set of weights. When α is small, reaching the bottom will certainly happen but after a large number of steps. When alpha is too big, it is possible for the steps to overshoot the minimum and zig-zag from side to side without ever settling at the bottom. One intuitive improvement to the learning rate dilemma is

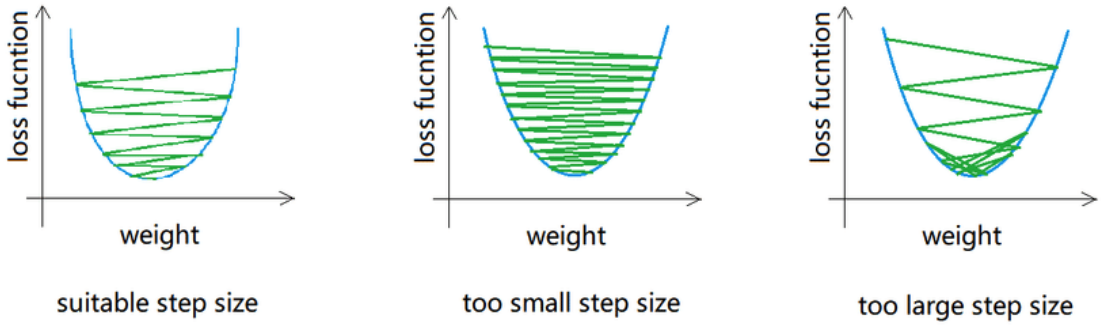


Fig. 8.5: Illustrates in a two-dimensional case, the tradeoff in choosing the right step size. Each step hops across the bowl to get closer to the bottom. The hop magnitude is based on the step size.

to employ a variable learning rate. One that maybe starts large, and shrinks down as the convergence approaches. This concept is known as momentum and in almost all cases is worth using. Momentum adds a second parameter, γ to SGD such that α decreases at a rate proportional to γ with each iteration. Different flavors of SGD use momentum and other tricks to effectively speed up convergence without overshooting it. As with most neural net parameter choices, the best SGD variant and best parameters are very much dependent on the problem and the dataset. Therefore it is often best to try a few and stick with one that seems to train fast while not sacrificing accuracy.

Plotting training curves is a common way to benchmark various neural net parameter choices as it conveys the training speed, as well as ability to learn. In a training curve, the loss is typically the y axis and the iteration (proportional to training time) is the x axis. Over time, the loss is expected to converge to some limit, hopefully close to 0 or at least lower than where it started. Loss typically decays exponentially as the weights are adjusted to be in the ballpark, but then subtle tweaks continue to widdle down the loss. Eventually, the loss somewhat flat-lines signifying the model's weights have converged and the model is no longer learning and probably overfitting. Figure 8.6 shows the most popular optimizers applied to the same dataset and shows how choosing the right optimizer speeds up training significantly.

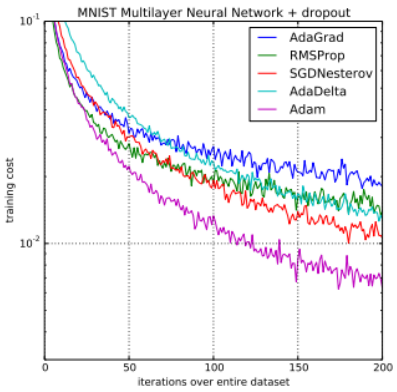


Fig. 8.6: Various common optimizers applied to the same dataset and trained for 200 iterations [41].

Adam, the superior optimizer in the illustration, adapts an intelligent algorithm for adjusting the learning rate and is often preferred [41]. Note that if the x axis were extrapolated to infinity, it is likely all optimizers would eventually converge to very similar weights. Therefore, the choice of the best optimizer is mainly a convenience of training time rather than accuracy, assuming the optimizer is configured to avoid overshooting the optimal weights.

Requlariza tion

Similar to classical linear models, it is possible to regularize the weights in the NN. This effectively controls the capacity of the networks and helps prevent overfitting. Usually, regularizers are applied on a layer by layer basis, where each layer may be regulated in some way, or not at all. L1 and L2 normalization can be applied to the weights just as in Classical Machine Learning and in effect prioritizes productive weights over weights that do not seem to impact the big picture prediction. In L2, weights are often linearly decayed to approach 0, while L1 will explicitly remove weights by permanently setting them to 0. Usually, L2 is preferred as it is more flexible.

Dropout is an extremely effective, simple and recently introduced regularization technique that complements the other regularization methods. While training, dropout randomly disables neurons based on a probability, p for a single iteration [80]. This essentially forces the NN to learn to perform without relying too heavily on specific neurons, or weights and thus prevents overfitting. This work utilizes dropout on nearly every hidden layer in the proposed networks.

Batch Normalization

A technique called Batch Normalization (BN) at a high level helps neural networks initialize by explicitly forcing the activations throughout a network to take on a unit Gaussian distribution at the beginning of the training. It has become standard practice to perform BN right before the activation function of each layer and it is done in this work as well [37].

Conclusion

In summary, training a neural net boils down to methodically passing the input through a series of layers which contain neurons tuned to extract specific types of features or patterns useful for solving the problem. These neurons are tuned by their weights which are updated via backpropagation with each training iteration in a way that ideally reduces the loss or prediction error over time. The optimizer dictates how aggressively the weights are updated and there are parameters and trade-offs to consider when configuring the optimizer. The next section includes a deeper dive into what a single neuron does in order to generate an output from multiple inputs.

8.3.2 Artificial Neurons

Recall from Classical Machine Learning a multidimensional linear regressor applies weights W to an input X such that: $z = f(X) = W^T X$, where the bias term is baked into the weights and z is shorthand for the linear operation. Neurons use this same linear function to apply weights to their inputs to obtain an output, but they often are configured to apply supplemental non-linear operations to z, which are called *activation functions*. The name bias, taken from transistor jargon, is similar to the y intercept in linear curve fitting. Another way to think about bias is that it is used to set the default behavior of the neuron (fire a 0 or 1), irrespective of the weights. A high bias makes the neuron require a larger input to output a 1, and a lower one conversely makes it easier. After training, each neuron has tuned weights which convey the relative importance of each separate input, and a single tuned bias value which dictates the neurons default behavior.

The activation function is analogous to the rate of action potential firing in the brain. The activation function starts with the weighted sum, z, then transforms it once more usually in a non-linear manner. Many activation functions have been proposed and common ones are shown in Figure 8.7. One important rule is for activation functions to be differentiable. In fact, any operation performed in a multi-layered NN must be differentiable. This requirement is fundamental in allowing the NN weights to be updated after a training iteration as the update rule relies on gradient descent which requires that every neuron based transformation be differentiated. The two activation functions used in this work will be described in detail: sigmoid and rectified linear unit (ReLU). Also note that the simplest activation function, a linear activation, simply outputs z as is.

Sigmoid

Historically the sigmoid, also called the logistic function, is the oldest and most popular activation function. It is named after it's "S" shape and it is clear from Figure 8.7 that the sigmoid acts as a sort of "squashing" function, condensing the previously unbounded output, z to the range 0 to 1. When z outputs a 0, the sigmoid maps that to 0.5. Infinitely large z outputs are saturated to 1 and infinitely small, 0, thanks to the e^{-z} term. Sigmoid activation is used extensively in binary classification as it conveniently outputs a probability between 0 and 1 as an output. Note, the softmax activation is a generalized variant of sigmoid to support outputting probabilities in a multi-class classification task. The sigmoid and softmax functions are also used to convert a classical linear regression to logistic regression for classification explained in the Classical Machine Learning section. Similarly, the Tan and ArcTan functions are also used for their sigmoidal shape, although these functions bottom out at or near -1.

Sigmoid activations were the basis of most neural networks for decades, but in recent years, with the advent of DNNs, they have fallen out of favor for layers other than the final output. The reason for this is when many layers with sigmoids are stacked and differentiated in order to update the weights, the gradient result tends to be very small. The magnitude of the gradients is proportional to the change in the weights, so a small gradient means the weights hardly update after a training iteration, i.e., the NN doesn't learn. This problem is widely known as the vanishing gradient problem. One of the breakthroughs in ALexNet was to replace sigmoid with ReLU as an activation function for most layers which effectively solves the vanishing gradient problem and allows DNN's to be trained effectively.

Rctified Linear Unit (ReLU)

ReLU's also borrow vernacular from the world of semiconductors as "rectified" is commonly used to describe diode behavior and a ReLU behaves very similarly to an ideal diode [64]. ReLUs let all positive values pass through unchanged, but sets any negative value to 0. This treats the vanishing gradient problem as the output of a ReLU is only bounded on the low end, so the gradient has room to breathe. ReLU is also incredibly efficient to compute as there is no arithmetic that needs to be done, simply a single $max(0, z)$ comparison. There are several improvements to the ReLU that do marginally better at the cost of sacrificing some computational efficiency. These improvements, PReLU and ELU along with the original ReLU are shown in Figure 8.7. The main benefit of the ReLU variants is to allow more freedom in the lower values z outputs as they are not set to 0. Similar to loss functions, there is very little theory to support choosing an optimal activation function and preference is instead based on empirical results from various datasets. In this work, ELU was found to train better, although slightly slower, so it is used instead.

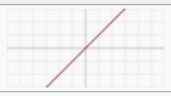
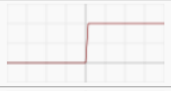
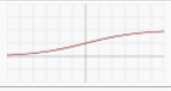
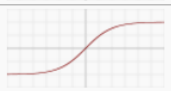
Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1-f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2+1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU)		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU)		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$

Fig. 8.7: Common activation functions as well as their derivatives where the activation function is expressed as $f(x)$ rather than $f(z)$. Note, some of them limit the min or max input value and all are differentiable

Conclusion

In summary, an artificial neuron on its own is not all that complicated. It simply applies a differentiable transform to the standard linear weighted sum of the inputs to obtain a single output. In other words, a series of inputs come from the neurons of the previous layer, the weighted sum of them expresses them as a single number, and the activation function morphs this number in a non-linear way. These differentiable transforms, known as activation functions, have different use cases and properties. Some such as sigmoids, conveniently output a probability while others such as ReLU serve as solutions to problems that prevent DNNs from scaling. Every training iteration, the weights, and bias of the neuron are adjusted depending on the error gradient resulting from gradient descent.

8.3.3 Architectures

Now that all the building blocks of neural nets have been established, it is possible to discuss the architectures which piece together neurons and layers to best expose the power of neural networks. It will be shown that combining multiple architectures and building blocks typically result in the best performing NN models. This section wraps up the neural network

and machine learning background and aims to bring together everything discussed related to neural nets so that it can be applied to real-world problem-solving.

Perceptron

The perceptron is the first version of a neural net and essentially resembles a single neuron with a step function for the activation function which either outputs a 1 or 0. Perceptrons have a much simpler update rule and do not require backpropagation, nor a differentiable activation function. Modern artificial neurons, while mostly the same as perceptrons, have more complicated update functions and support other differentiable activation functions such as sigmoid or ReLU.

Fully Connected

A fully connected network (FCN) defines the type of multi-layered network that has been discussed so far. FCNs consist of N hidden, fully connected layers where each layer has a specified number of nodes or neurons as a parameter. The FCN can be considered deep when $N > 2$, although "deep" is more of a marketing term rather than a prescribed property. It is typical for the hidden layers to either have the same number of nodes in each layer (as in Figure 8.8) or for them to decrease (i.e., halve) each layer forming a pyramid-like structure. A FCN by itself is not a very powerful feature extractor and is typically used with manually extracted features. It is also very common to use a FCN as the last layer or two of a more complex architecture such as those listed below.

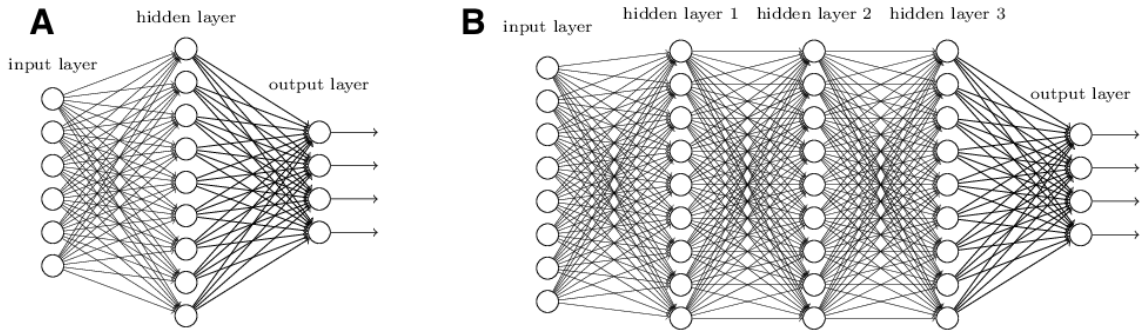


Fig. 8.8: Fully connected neural network examples showing a shallow single layer (A) and a "deep" three layer variant (B)

8.3.4 Convolutional Neural Networks

Convolutional neural nets (CNN), sometimes called conv nets, have become standard for automatic feature extraction in images and spectrograms as they scale and perform better than FCNs. The reason: as the size of the input image increases, the number of FCN weights needed at the input layer increases substantially as well. This creates a bloated, inefficient

and prone to overfitting FCN. CNNs customize the architecture to support images in a practical way. The standard architecture is inspired by LeNet which was first proposed in the 90s and also uses many concepts from AlexNet [48, 43]. CNNs are as complicated as they are powerful, so several new concepts will be introduced in this section so the architecture can be understood.

First, recall an image is a 3D matrix consisting of a depth equal to 3 channels (RGB) with each channel representing a matrix matching the pixel dimensions of the image. The shape of a normal color image is thus: *(height x width x 3)*. A spectrogram only has a depth of one which defines the magnitude of each pixel and has a shape of: *(frequency x time x 1)*. When discussing CNNs it is easier to think of the layers as 3D cubic volumes, or block layers rather than 2D matrices or 1D vectors. CNNs introduce another type of layer called a convolutional layer (conv layer for short) that differs from a fully connected layer in that it handles inputs and outputs in the form of volumes rather than vectors.

CNNs are in some ways influenced by the human brain's visual cortex in the sense they both extract patterns from images in order to perceive what is in the image. The dog breed detection example in the last section describes the high-level nature of how successive conv layers extract features, not far off from the visual cortex. As stated in the dog example, the extracted features start basic and become more complex as they approach the final layer. This is a byproduct of the standard CNN architecture. The architecture typically starts with a large, but thin (depth wise) input layer, such as an image with shape *(500 x 500 x 3)*. Each successive conv layer applies various filters in order to extract relevant information. This elongates the depth dimension as several alternative representations of the input are generated through convolving multiple trainable filter kernels with the input and passing the result through an activation function like ReLU. Following a conv layer, a pooling layer reduces the h x w dimensions to make up for the increase in depth and keep the total volume from getting too large. Pooling will not be covered further, but it essentially downscales h x w by taking the max or average of small patches distributed across the image. Usually, there are multiple conv + pool layers which continue to reduce h x w and increase depth. The last conv layer might have a shape of *(5 x 5 x 300)* and rather than representing the original *(500 x 500 x 3)* pixels in the image, it now describes 300 *(5 x 5)* image feature maps. Each feature map represents an image-based feature the CNN has learned to extract through its tuned image filters (also called kernels).

Following the last conv layer, it is common to perform the classification or regression step on these automatically extracted feature maps using one or more fully connected layers. This is essentially done by attaching an FCN onto the end of the CNN. More specifically, the 3D conv layer is flattened to a 1D vector using some form of averaging or reshaping, then treated as the input layer for the FCN. An example CNN architecture diagram is shown in Figure 8.9.

It is typical for the conv layers to form a pyramid-like structure as they grow in depth and shrink h x w.

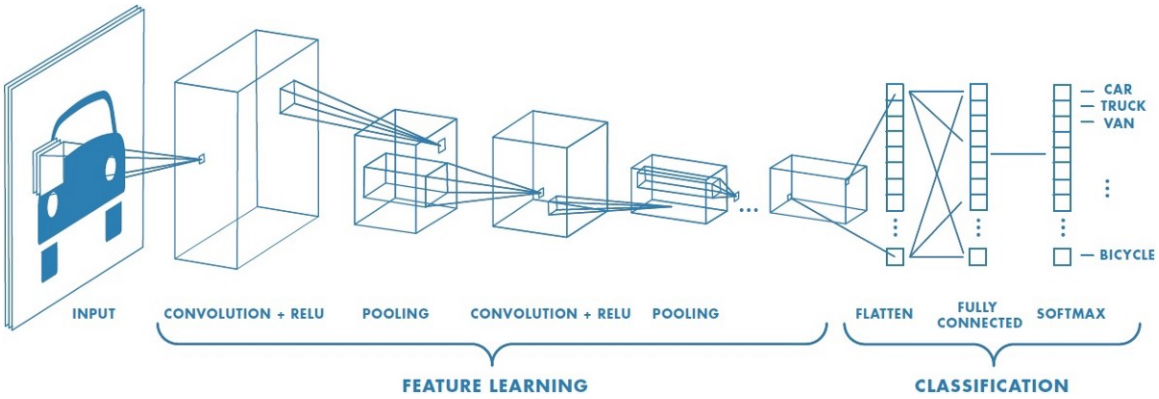


Fig. 8.9: Standard convolutional neural net for image feature extraction in order to classify objects

The process of "convolving multiple trainable filter kernels" mentioned above is what does the heavy lifting in a conv layer, so it will be explained more thoroughly. When configuring a conv layer, a few parameters must specified: the input volume shape (h x w x d_{in}), the desired output depth, d_{out} (single number) and the filter kernel size (k_1 x k_2). The input shape is defined by the previous layer, but d_{out} and (k_1 x k_2) are configurable parameters. The depth specifies how many feature maps are created from the input, usually between 16 and 512. The filter size is typically a small square ($k = k_1 = k_2$) and k is usually odd to ensure there is a center pixel. Usually filters sizes range from $3 < k < 13$. Every filter is small spatially (along width and height), but extends through the full depth of the input volume, therefore the depth part of the filter is not configurable as it must match the d_{in}. To obtain the output volume of the conv layer, a total of d_{out} filters are convolved with the input volume to create a total of d_{out} 2D feature maps which usually have the same h x w as the input. The feature map represents the response of the filter kernel applied to each spatial position of the input and each pixel in the feature map functionally represents a weighted combination of the spatially equivalent pixel in the input image, as well as the neighboring pixels. More neighboring pixels are considered as k is increased. An example of the feature map created from convolving a filter with an input image is shown in Figure 8.10.

In a FCN, each neuron receives information from every neuron in the previous layer. Given the much larger volume of CNNs, it is impractical to connect neurons to all neurons in the previous volume. For this reason, each neuron has a receptive field that is the size of k which narrows its focus to a specific spatial region of the input volume, rather than the whole thing. There is a neuron for each spatial pixel in generated feature maps. The specific way backpropagation updates the filter weights will not be covered as it involves complicated details to make the process efficient. Essentially, for a given conv layer, with d feature maps there are k*k weights required for each feature map, so d*k*k weights per

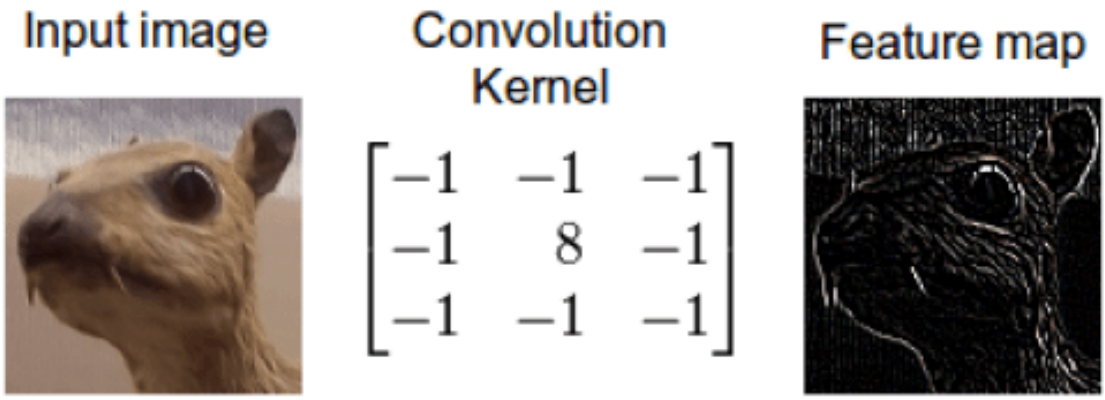


Fig. 8.10: Example showing the feature map that results from convolving a filter kernel with an input image. The 3x3 filter kernel effectively extracts the edges of the input when applied

conv layer. Therefore after training the CNN, each layer has d trained filters that each extract a different feature map from the input volume.

An example showing how the complexity of the features evolve as the layers get progressively deeper is illustrated in Figure 8.11. These features were trained for facial recognition using a Convolutional Deep Belief Network which is a specific variant of a CNN [49]. The final extracted features which are passed to the FCN at the end of the network capture incredibly specific details such as facial hair, nose to eye ratios and facial hair. The CNN effectively expresses an input face as a weighted combination of each feature in the final layer and passes that to the FCN which must simply connect this weighted combination with the associated label, or in this case identity. Clearly, the CNN is doing most of the work. So far all

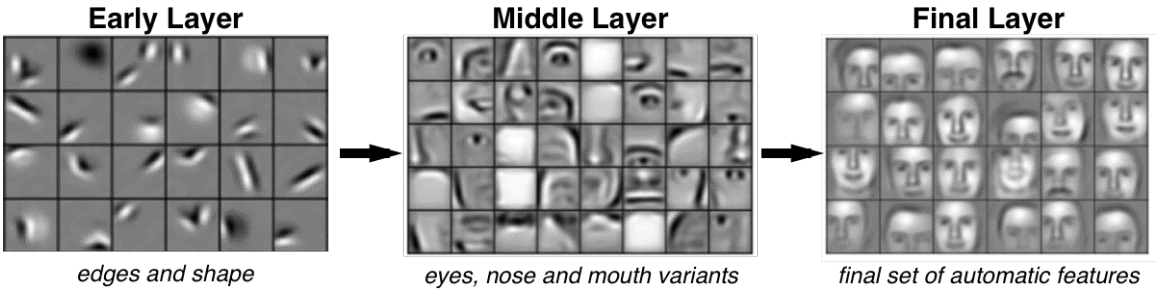


Fig. 8.11: Types of features extracted at various layers in a convolutional neural net trained on faces.

examples have been related to image input, but as mentioned, other 3D inputs can be used, such as radar or sound data expressed as a spectrogram. Using alternative inputs is much less common, and still being actively researched. Unlike images, spectrogram dimensions have meaning more significant than height and width, namely one dimension represents time, and the other frequency. As covered in the Sound Background chapter, a great deal can be interpreted from a spectrogram, and CNNs certainly have the power to extract whatever it finds useful. While the extracted feature maps may be less intuitive than the faces in

Figure 8.11, they have been shown to perform well at tasks such as sound classification and this work will show the effectiveness of CNNs for extracting flow from sound [76]. Usually, the trained filters expose the existence (or lack thereof) of certain impulsive sounds or background, drone sounds. There are some nonideal properties of CNNs that prevent them from understanding higher level patterns within a spectrogram that humans can easily spot, but when the sound is not rhythmic this is less of a problem. It will be shown in the next section, that there are better architectures to use when the input is a time series with an embedded pattern.

CNNs biggest limitation is that it tends to create a black box model. Unlike classical machine learning where the relative impact or importance of the input features can be quantified, in CNNs and NNs in general, it is difficult to know which features are being extracted and how they are being used. This is the downside to employing millions of neurons to solve a problem. There are attempts to explain a CNN model by looking at the feature maps, as shown above, but it is still largely a research topic and not very well understood. There are several examples where the NN learns to cheat by exposing a flaw in the data rather than solving the actual problem. In this work, for example, a NN was found to be remarkably accurate at tracking the speed of fan powered airflow. Unfortunately, it turned out to be using the pitch of the humming, or resonance frequency as the predictor for speed, not the actual sound of airflow. This was quickly realized when it utterly failed to work on a different fan or human-powered airflow. As Uncle Ben advised Spiderman, "With great power comes great responsibility", and it is unfortunately up to the engineer to be responsible, at least until an AI is built to handle that part too.

CNN Conclusion

As mentioned earlier, the 2012 explosion of DNNs can be attributed to a massive influx of image data, AlexNet, and powerful, ubiquitous GPUs. AlexNet was essentially the CNN architecture described in this section with a pyramid-shaped stacking of conv layers. The GPUs were necessary as they are optimized to perform filter convolutions on images (this is required in video game rendering too). Combining GPUs was the key to AlexNet as it allowed very large volumes to be processed at each layer which yielded a massive amount of features capable of achieving a miraculously low error rate on the substantial ImageNet dataset. From then on, the research in CNNs has exploded and has recently found its way into products such as photo apps and autonomous cars. The next architecture, Recurrent Neural Networks (RNN), has undergone a similar explosion in the last decade but for different reasons than the CNN. The RNN is less relevant in this work, so it will only be covered from a broad perspective.

8.3.5 Recurrent Neural Networks

Recurrent neural networks (RNNs) have been a staple for sequential data processing such as language translation or speech recognition. In a traditional neural network, the assumption is that all inputs (and outputs) are independent of each other, but for many tasks, that assumption is very limiting. When predicting the next word in a sentence it is essential to know which words came before it, not just the last word, but the last few. RNNs are called recurrent because they perform the same task for every element of a sequence and each successive output is dependent on the previous computations.

Back to the brain analogy, RNNs have "memory" which captures information about what has been observed so far. Like the human brain, the memory is not unlimited due to storage capacity, so there must be a decision rule for deciding what to remember and what to purge so new information can replace it. A few RNN variants have become popular due to the way they handle old and new memories. RNNs are similar in concept to a memory cell, which has gates that allow new memories in and to reset or clear old memories. The two most popular variants are the Long Short Term Memory (LSTM) and the Gated Recurrent Unit (GRU) [33, 14]. Unmodified RNNs are also particularly susceptible to the vanishing gradient problem covered earlier, which is alleviated by these variants because they constantly update and reset their memory preventing it from falling into a stagnant state. A typical RNN structure is shown in Figure 8.12. The main takeaway is the previous inputs are stored and used to make predictions for future inputs.

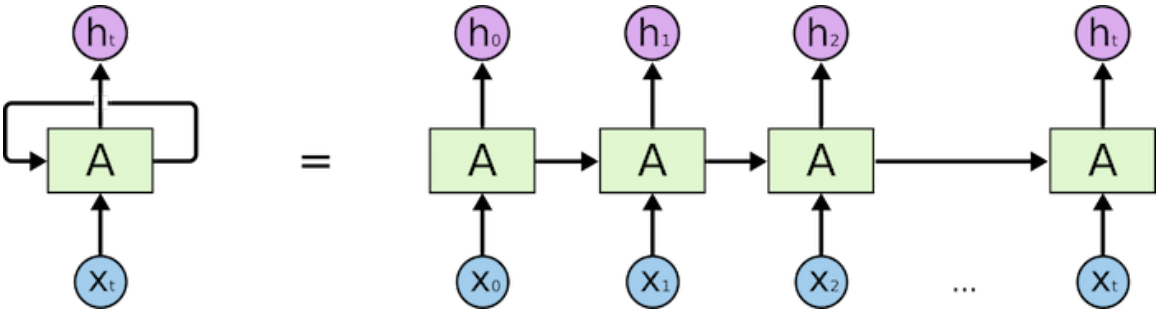
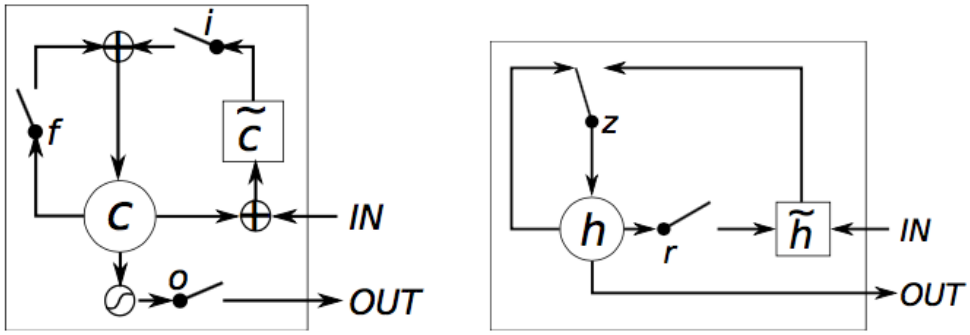


Fig. 8.12: Standard recurrent neural network architecture, when unfolded it reveals the previous states can be recalled, the labels are not important for this example.

The key difference between a GRU and an LSTM is that a GRU has two gates (reset and update gates) whereas an LSTM has three gates (input, output and forget gates). The newer GRU unit controls the flow of information like the LSTM unit, but without having to use a memory unit. It exposes the full hidden content without any control. GRU's are generally preferred as they are usually on par with LSTMS, but are simpler to understand and perform better with less data. LSTMs should, in theory, remember longer sequences than GRUs and outperform them in tasks requiring modeling long-distance relations, but it seems empirically

this does not hold up. Figure 8.13, shows the different structure of LSTM and GRU cells. The details are presented in the source paper and will not be covered here.



(a) Long Short-Term Memory (b) Gated Recurrent Unit

Source: GRU (Cho 2014) (a) LSTM and (b) gated recurrent units. (a) i, f and o are the input, forget and output gates, respectively. c and $\tilde{c}$ denote the memory cell and the new memory cell content. (b) r and z are the reset and update gates, and h and $\tilde{h}$ are the activation and the candidate activation.

Fig. 8.13: Comparison of Long Short Term Memory and Gated Recurrent Unit cells

RNNs and their powerful variants have been shown to successfully understand, translate and generate written text. They have also demonstrated an ability to label and caption images and recognize recorded speech audio. Most personal assistants like Amazon's Alexa, rely on some form of an RNN to listen to a request, then reply with synthetic speech. When images and audio are the input, it is typical to use a CNN to extract features, then an RNN in place of an FCN to process the CNN features in a sequence order. Using an RNN in place of a FCN for sound has an added benefit of learning how the amplitude and frequency varies across time which is incredibly useful for real-time or rhythmic sound. This combination is usually referred to as a convolution recurrent neural network (CRNN) and it is explored in this work as a potential solution for airflow to sound.

Conclusion

The architectures presented are merely scratching the surface in terms of the multitude of approaches artificial neurons can be arranged to solve problems. The depth of the concepts covered is somewhat related to its utility in this research. In summary, FCNs are multi-layered basic neural nets and are ideal when manual features have been extracted, or are simple to extract. CNNs have the ability to automatically extract complex information from 3D input volumes. They are often fed into FCNs which then perform the classification or regression steps on the automatically extracted features. CNNs are limited when it comes to sequence data, so RNNs are often employed as they possess the ability to remember patterns and information from previous data. The feature extraction of a CNN can be combined with the pattern recognition of an RNN to create a powerful hybrid architecture known as a CRNN.

8.4 Conclusion

This chapter has been a whirlwind of approaches designed to learn predictive models from copious amounts of data. As was shown in Airflow Physics, physics models fall apart when introduced to the complicated non-ideal properties of reality. Since machine learning relies on data from reality, it has a competitive edge in terms of learning ways of representing the physical properties observed through the data. The downside is, they typically require tens of thousands of data points to learn the necessary relationships.

Machine learning can be broken up into a few categories based on whether or not labeled data exists. In general, supervised methods require labeled data to learn a transform function from the input data to the output ground truth. Unsupervised learning is used when data is unlabeled but may have high-level properties that can be exploited via clustering. The types of models explored were broken into *classical* and *neural network* categories where the classical methods require manually extracted features and have been used in practice for decades. On the other hand, deep neural networks recently rose to fame due to the advent of GPU computing and the existence of extensive datasets. It was shown that DNNs, specifically CNNs have the powerful ability to automatically extract features from input data, but it is difficult to know what the extracted features represent and which have the most impact on the final prediction. For this reason, both classical models and NNs are evaluated in this work. The intuition is if the CNN does remarkably better, then the manual features are not painting the complete picture. Contrary, if the manual features exceed the NN performance, then the NN method is not doing a good job at extracting features.

If the hope was to read this chapter and walk away with the "best" way to apply machine learning for a problem, then the conclusion is probably fairly disappointing. In machine learning, there's something called the "No Free Lunch" theorem. In a nutshell, it states that no one algorithm works best for every problem. This is especially relevant for supervised learning. There are clues of when to use one versus another based on the type of input data or intended model, but in general, the only way to know is to train them all and evaluate them against each other using a test set.

The following chapters will dive into applying this "machine learning bake-off" to the spirometry sound dataset with the goal of providing spirometry metrics and curves from sound input. The Dataset chapter will introduce the dataset as well as some of the limitations inherent to it, the Methods chapter will outline the proposed machine learning methods and manual features utilized, and finally the Results chapter will reveal the results of the bake-off as well as important insights and observations.

9 Related Work

The related work is broken into four sections. First a general overview of similar mobile health work is presented, followed by a focus on smartphone based spirometry. The third section is a broad overview of ultrasonic and infrasonic airflow measurement approaches and the chapter is concluded with an overview of related sound-based deep learning research. The bulk of the related work stems from the ubiquitous computing, pulmonary health, acoustical modeling and deep learning communities.

9.1 Mobile Health

The mass adoption of smartphones has led to several innovations related to mobile health (mHealth). A subset of mHealth research focuses on medical record input and storage in the form of apps with built-in calculators and access to shared medical databases. Unfortunately, these services have yet to see widespread use outside of academia [15]. Another active topic in mHealth which has made it into the commercial world is activity tracking, including fitness and sleep services. Some of these exist in the form of external wearable hardware that may interface with a smartphone such as Fitbit, Nike+, and most other smartwatches, while others, such as Google Fit and Apple Health make use of the accelerometer and GPS within a smartphone to count steps and track exercise and sleep. These activity tracking products typically do not claim to offer any clinically relevant information and are therefore not suitable for clinical use.

Another class of mHealth research focuses on the personal measurement of vital signs and monitoring of chronic diseases using a smartphone. These are somewhere between non-clinical activity tracking and gold standard medical devices. They often are presented as proof of concept ideas that are evaluated on a sample size smaller than 100 patients. Many non-invasive blood screening apps utilize the camera and accelerometer to measure pulse, blood pressure, blood oxygens levels, hemoglobin concentration and arterial stiffness [63, 89, 88]. Other vision based apps can screen for exterior health conditions such as jaundice, melanoma and diabetic wounds [17, 56, 86, 90]. Additionally, a number of apps use the microphone as a sensor to track respiratory rate, sleep duration, coughing and sneezing [65, 47]. The majority of these health monitoring apps provide a non-invasive, portable supplement or replacement to a traditional medical device using only a smartphone. The

focus of this work is in a similar vein as it aims to sense pulmonary functionality similar to a spirometer using the microphone of a smartphone. While the approach is novel, the idea has been floating around in academia for over a decade, as the next section will show.

9.2 Spirometry via Sound

There is a rich history of spirometry products as outlined in Section 4.1, but the prior work most relevant to this research uses sound as a proxy for airflow and utilizes a smartphone to record the sound and process the signal. Early publications demonstrating sound-based smartphone spirometry began emerging around 2011 but required either an external microphone placed at a fixed distance or an external breathing tube [1, 44]. Nonetheless, this work contributed signal processing techniques for isolating the airflow sound by applying voice activity detection (VAD) techniques and tracking sound energy versus time. Furthermore, the early work introduced methods for detecting and classifying common spirometry errors [29].

Shortly after, in 2012, Larson et al. published SpiroSmart, a sound based spirometry solution that uses an iPhone microphone and cloud-based processing for portable, accessory-free, spirometry testing [46]. SpiroSmart is capable of outputting FEV1, FVC, and PEF but did not specifically evaluate the output of spirometry curves such as flow volume. It achieved an average error of 4.9% on FEV1, but was only evaluated on 52 patients and had significant error on the small sample of unhealthy, low FEV1 patients. Other than demonstrating smartphone sound based spirometry is indeed possible, this work also contributed crucial signal processing techniques and physical models for measuring the airflow via smartphone microphone. Furthermore, it spawned the global data collection effort that resulted in the dataset used in this work.

Since the genesis of SpiroSmart, many other versions have been proposed with the same objective but different processing techniques as the solutions. These variants are evaluated on less than 50 healthy college student participants, so the error metrics must be taken with a grain of salt. In 2013, Xu et al. designed and developed an Android mobile phone application for lung function diagnosis called mCOPD which measures FVC and FEV1 using the microphone. They evaluated mCOPD on 40 patients and obtained an FEV1 percent error of 6.1%. The processing algorithm was much simpler than SpiroSmart as it was simply a linear mapping from sound energy to wind speed from which FEV1 and FVC are derived [92]. Recently, in 2017, Zubaydi et al. demonstrated an Android-based COPD management app very similar to mCOPD [95]. It utilizes a tuned quadratic formula to converts sound energy versus time into flow versus time and achieves an FEV1 average percent error of 3.5% on 25 subjects. So far the work that has followed SpiroSmart has used smaller, less diverse

sample sizes while offering minimal improvements error wise. The processing algorithms, however, are much more straightforward and intuitive.

SpiroCall, published in 2016 by Goel et al., while a direct descendant to SpiroSmart, offers a fresh twist to mobile spirometry by enabling the transmission of spirometry efforts over a standard cell phone line rather than an internet enabled smartphone. This effectively enables a much needed portable spirometry solution in developing countries. This work evaluated the technique on several phones and also introduces a vortex whistle which converts the airflow into sound, making it easier to capture and process the spirometry effort. They evaluated the proposed methods on a good balance of 50 patients and achieved around 8% FEV1 percent error without a whistle and around 5% with it. They also demonstrated the whistle was capable of producing accurate flow versus time curve, although did not sufficiently evaluate it.

While the main objective of this work as well as the work mentioned in this section so far is to predict spirometry metrics, there has also been substantial work around classifying spirometry errors, both from the flow volume curve and from the sound itself. In 2014, Melia et al. demonstrated an automated feature-based method for detecting four common spirometry errors, this work was recently improved on by Luo et al. which used a subset of the dataset used in this work [59, 54]. Finally, early results of the Confidence model outlined later in this work are being published by Viswanath et al., which show spirometry error detection is possible from just the sound of the exhale.

9.3 Airflow via Inaudible Sound

Another goal of this work is to explore airflow sensing using infrasound and ultrasound in addition to audible sound. Unlike the work of the last section, the work in this section takes more of a physics-based approach to modeling as opposed to data-driven. The physical modeling is covered in depth in Section 7.2.

Much of the prior work in measuring wind speed relies on multiple microphones such that differences in phase can be mapped to wind speed [8, 26]. Other work, dating back to the 1970's leverages the infrasonic vibrations caused by natural wind flow to measure the velocity [87, 58, 11]. Raine et al. demonstrated it is possible to measure air velocity in an air conditioning duct using a specialized ultrasonic setup [71]. Ultrasound is also utilized in active research for breathing monitoring, specifically for sleeping subjects when motion is assumed to be minimal [4]. This work mostly tracks the chest and mouth movements rather than the airflow so it does not investigate the velocity of the air. None of these infrasonic

and ultrasonic applications, however, have been shown to work on a smartphone as they require specialized, calibrated hardware.

9.4 Deep Learning

The most promising methods in this work stem from recent advancements in deep learning, specifically in the sound domain. Unlike vision based deep learning, sound is not nearly as well explored outside of speech, although it is gaining significant momentum. General deep learning advances and pivotal work is covered in depth in Section 8.3.

The publishing of the 8k urban sound dataset by Salamon et al. spawned several innovative neural network architectures designed to classify and understand sound, including the deep learning CNN based method later published by the dataset creators [77, 76]. Parascandolo et al. build on this concept and offer accuracy improvements by including recurrent layers in addition to the feature extracting convolutional layers to extract time-varying patterns, known as a CRNN [69]. More recent work has shown a custom type of convolutional layer known as a "gated" variant can outperform standard convolutional layers in sound-based feature extraction [93]. All of these methods are explored in this work, although no prior deep learning work is focused on airflow or spirometry was discovered.

This year, Google published a massive Youtube based labeled audio dataset known as Audioset which has over 500 different labels, including coughs, sneezes and throat clearing [23]. This has spawned new conferences and competitions around sound detection and will no doubt result in groundbreaking innovation similar to the effect ImageNet had on vision-based detection.

10 Experiments

> *No amount of experimentation can ever prove me right; a single experiment can prove me wrong.*
>
> **— Albert Einstein**

In order to support and validate the research and ideas presented in this thesis, a number of experiments were performed. The experiments are briefly documented and listed in chronological order.

10.1 Airflow

The following preliminary experiments on airflow and and airflow measurement with a MEMs microphone provided important insights into the difficulty of building a physics-based model.

10.1.1 Constant Airflow

The first experiment, conducted in the context of airflow physical modeling, explores using linear models to track constant airflow. The setup involved a computer fan and the speed was modulated simply by controlling the input voltage to the fan. The airflow at each voltage setting was measured 1 foot away using a hand-held anemometer. The voltage speed mapping was indeed linear, which confirmed modulating the voltage had the same effect as modulating the airflow in close proximity. This relationship is shown in Figure A 10.1. The measured airflow speed ranged from 1 to 2.5 m/s.

Following the confirmation that airspeed is linearly proportional to fan voltage, a smartphone was set up to record the sound of constant flow at each voltage setting. A simple algorithm was developed to average the sound into loudness bins, then a simple linear fit was computed to estimate the fan speed from the loudness bins in real time. The linear mapping is of the form:

$$V_{air} = m\bar{L} + b \tag{10.1}$$

where $\bar{L}$ is the average loudness of a small, 100ms audio segment and V_{air} is the velocity of air measured 1 foot away along the centerline of the fan in m/s. Empirically, m was computed to be 4.54 and b, 46.17. It was also found this model is relatively distance independent over the range of 0.5 - 2.0 feet, as long as the audio is normalized. This suggests normalization corrects for distance, assuming the SNR is high enough. The results from testing this transform are shown in Figure B of 10.1. The findings suggest constant airflow can be tracked via sound fairly easily assuming the phone, distance, and fan are consistent.

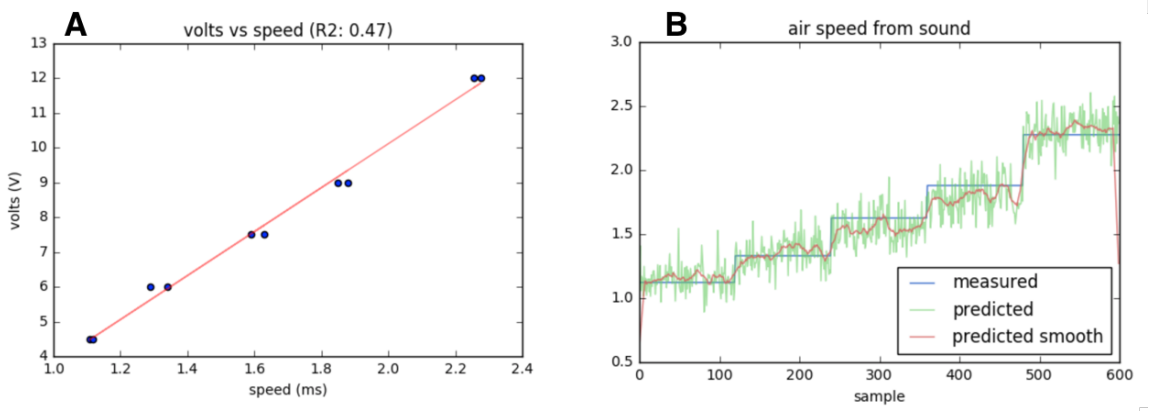


Fig. 10.1: Results of the constant airspeed from sound experiment. *A)* show the linear speed voltage mapping and *B)* shows the mapping works in real-time airspeed tracking

More complex machine learning models were trained on spectrogram representations, but the models clearly were learning the speed from the resonance frequency of the fan, which was clearly viable in the spectrogram. For this reason, only amplitude based methods were used to model fan airflow. This is one of many examples where neural networks learn to cheat rather than solve the problem in the intended way. Future efforts to use neural networks, were approached more carefully as to avoid this type of unexpected behavior.

10.1.2 Electro-Mechanical Lung

This experiment involves the design of a blower fan powered airflow generator with the purpose of supporting various data collection efforts. The design is inspired by the human respiratory system and built with components found at a local hardware store. It utilizes a standard four-wire pulse width modulation (PWM) controllable blower fan which is controlled using an Arduino with a custom, high PWM capable, firmware. The fan output is directed into an expandable mylar bag which acts as a poor substitute for the lung. The other end of the mylar bag is connected to a flow control valve which functions as an obstructive lung disorder proxy. The valve is connected to corrugated plastic tubing which is about the lengths and texture of a trachea. This 4 inch trachea substitute is connected to smoother tubing which is about 8 inches and emulates the rest of the respiratory piping up to the mouth orifice. Flow modeling efforts utilizing this device debunked the results of the first

constant airflow experiments as it showed the model fails when exposed to non-constant, or rapidly changing airflow. This result motived the transition from physics based models to machine learning models.

ATS Waveforms

The other purpose of this devices was to generate similar airflow curves to a $10000 American Thoracic Society (ATS) waveform generator. Sound and flow data were collected from a local ATS certified flow generator located at the Seattle Children's Hospital. This device is similar to what is used to certify spirometers and has a very accurate flow output, based on the human respiratory system. It can replicate the 26 ATS verified waveforms used for calibrating spirometers and has the capabilities of simulating obstructive and restrictive behavior and many other common patterns. The Arduino electro-mechanical lung is far less accurate and useful compared to the ATS flow generator, but it does provide similar functionality suitable for experimentation at significantly lower costs.

10.1.3 Ultrasonic Airflow

Prior work has confirmed it is feasible to measure hand motions via ultrasonic Doppler shift using only a laptop [30]. Additional work has also shown embedded hardware can be designed to measure the constant airflow in an air conditioning duct [71]. In an effort to combine these ideas and measure human airflow, a simple web-based app was developed to facilitate ultrasonic motion experimentation.

This app uses laptop's speakers to broadcast an inaudible 20kHz sine wave, which is then received by the laptop microphone. Any motions or fluctuations in the air manifest as subtle frequency shifts which can be picked up by the microphone as it can compare the broadcasted signal to the measured one. Using the Doppler shift principle, these frequency shifts can be transformed into relative motion vectors which have a magnitude and an angle.

The app successfully implements this theory and plots the motion magnitude versus time in real-time. The app is very useful as a learning tool and a data collection tool. It includes functions to tweak the ultrasonic frequency or play an audible sound that correlates to the motion magnitude, effectively turning a laptop into a motion-based Theremin style instrument. The app has been tested in the Chrome browser on a 15" Macbook pro but also works on other laptops with stereo speakers and a center-mounted microphone. Unfortunately, it does not work on a phone yet.

The code is open source so others can build ultrasonic functionality into their websites. A simple demo is accessible at (start with low volume): https://jake-g.github.io/spiro-doppler/, and an example of the user interface is shown in Figure 10.2.

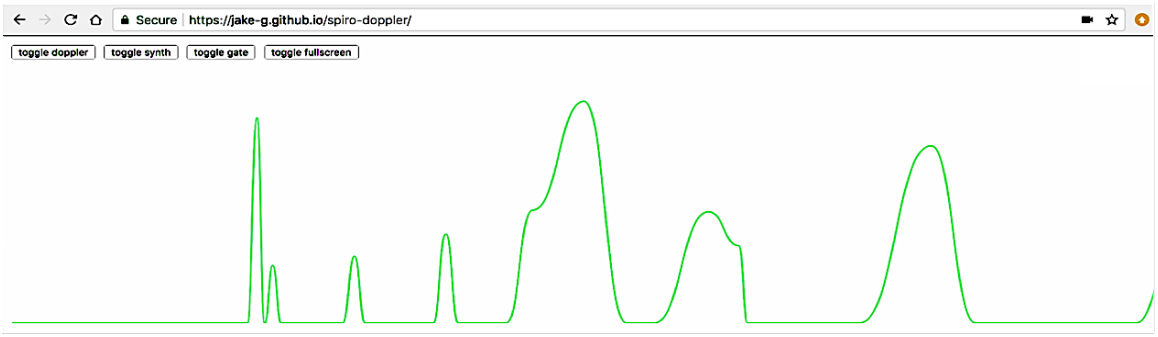


Fig. 10.2: A screenshot of the open source ultrasonic Doppler based motion magnitude visualizer

This app was originally built to support experimenting with this concept as a potential method for measuring forced exhale airflow on a smartphone as either a replacement or supplement to the audible sound methods covered in this work. Ultrasonic measurement is attractive as external audible noise will not affect the measurement. The downside is that motion is a form of noise in this sensing technique and difficult to control, especially if the user is holding the phone. Preliminary experiments show it may be possible to measure airflow, but it is difficult to tell whether the movement can be attributed to the airflow or the movements of the chest muscles or other sources. A more thorough investigation is required in order to validate this technique for use in spirometry.

10.2 Spirometry

These experiments were conducted to gain a better understanding about spirometry measurement techniques as well as provide software and tools for future experiments.

10.2.1 DIY $30 Spirometer

In an effort to learn more about airflow sensing and spirometry, as well as obtain a real-time exhale flow recorder, a DIY spirometer was designed, built and calibrated. The design requires a NXP differential pressure sensor ($10), a 16-bit ADC ($5) and an Arduino ($10). Disposable mouthpieces can be connected to the pressure sensor and the assembly can be used as a traditional spirometer. The prototype is shown in Figure A of 10.3 and has been open sourced.

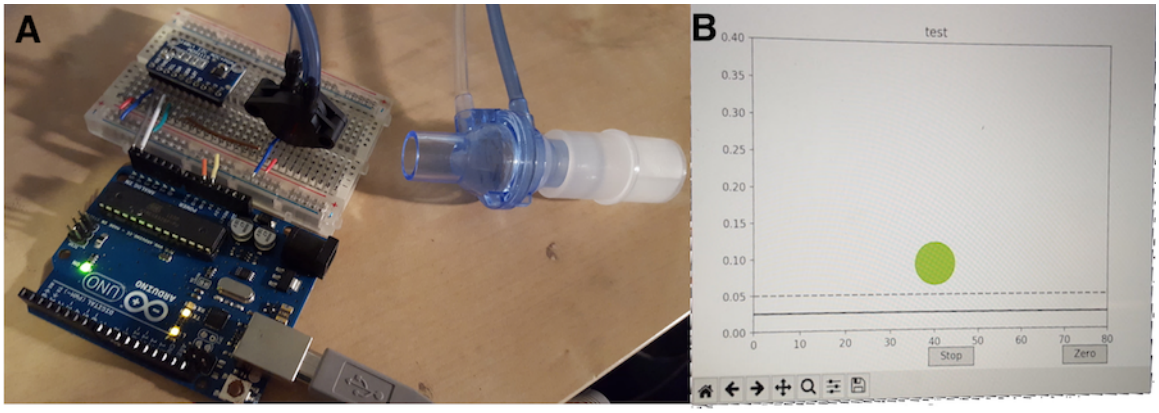


Fig. 10.3: (A) shows a prototype of the $30 DIY Spirometer and (B) is an example screenshot of the feedback tool for the Human Airflow study. The user is instructed to try to exhale such that the ball is centered on the horizontal line.

10.3 Deep Learning

While only the top models are evaluated in Chapter 13, there were hundreds of variants that didn't make the cut. Since January of 2018, a GPU machine has been running parameter tuning optimizations for the deep learning models. A simple genetic search algorithm is used to automatically adjust the architecture parameters on a weekly basis with the hopes of finding an architecture that works best. It is in a way using semi-supervised machine learning to optimize supervised machine learning. The process is mostly automated other than defining the range of the parameter sweep every week and checking the results. Simple power calculations estimate the electricity cost of training 24/7 since January amount to about $50 , or $5 per month. Figure 10.3 shows ten days worth of experimentation for the final refinement of CurveNet. Many of these converge to around the same loss by 200 iterations, but earlier in the process, the results were incredibly variable. At this point, in May, the architecture is fairly well defined and only small changes are made between training sessions. Note, the curves are smoothed and the faint silhouette allude to the much noisier source plots.

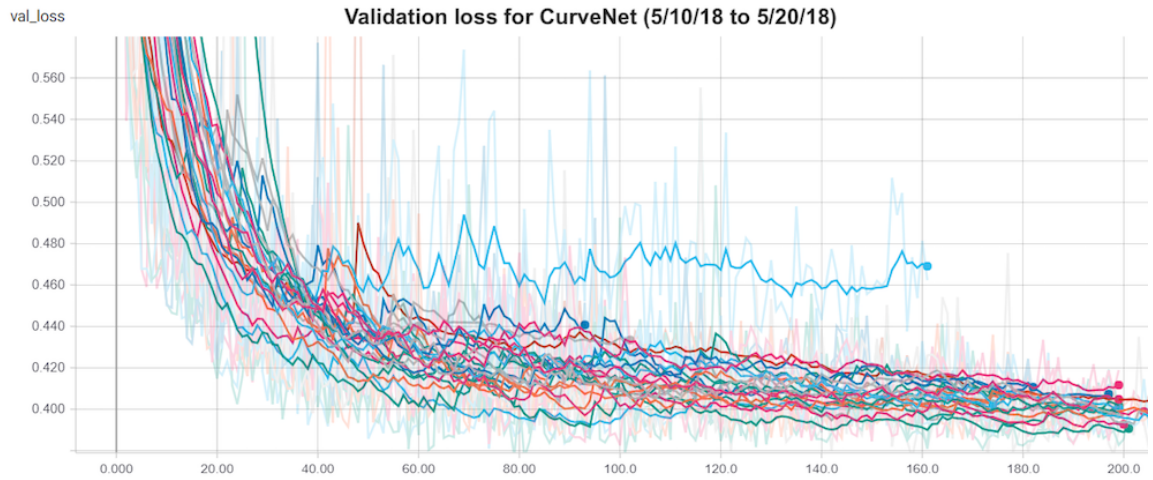


Fig. 10.4: Represents 10 days of CurveNet experiments. The plot shows the loss relative to training iterations. Most models converge to around the same point, but the speed and final accuracy varies.

11 Dataset

> *It is a capital mistake to theorize before one has data.*
>
> **— Sherlock Holmes**

In any machine learning problem, the key to building a generalized, accurate model lies in the dataset. Creating and organizing a dataset is often the most time consuming, tedious part of a data-driven problem, and often has huge impacts on the outcome. Even the best models are crippled by poorly organized or mislabeled data. This chapter describes the monotonous but important process of cleaning and organizing the data, which took up approximately one-third of the R&D time.

11.1 Collection

Motivation

The motivation for exploring machine learning methods to sound-based spirometry stemmed from the massive amount of existing exhale sounds and ground truth spirometer data. This data was collected in a joint effort between the original SpiroSmart authors, the Seattle Children Hospital and the Spirometry 360 organization. Through this collaboration, spirometry data was collected at several clinics in global locations such as Greece, Bangladesh, and Russia. In addition to collecting ground truth data, an iPhone app built to record the sound of a forced expiratory effort was deployed and used in conjunction with the traditional spirometry data collection procedure. The main goal of this global effort was to build a large dataset to be used for developing a sound-based spirometry model, among other things. Today, the dataset is comprised of approximately 40,000 entries from 8000 unique patients collected at numerous clinics. One of the main challenges in this research is unifying the data from various clinics with different protocols and timezones into one cohesive dataset suitable for machine learning. The following chapter outlines the data collection procedure and the steps taken to unify the dataset, along with the issues encountered along the way and finally, the key dataset statistics and observations.

Procedure

Data is collected in terms of sessions which is a collection of trials for a given patient. Sessions are generally separated by days or weeks. In a typical session, a trained clinician

first instructs and coaches the patient on the breathing maneuver and the patient is given an opportunity to practice. The instructions follow the best practices and guidelines for a spirometry maneuver outlined in Chapter 5 Spirometry. When ready, the patient performs the maneuver into an EasyOne Spirometer and the result is stored internally. Following the first successful trial, the patient is introduced to the SprioSmart data collection app and instructed to perform the exact same spirometry maneuver except into the smartphone. Some additional restrictions are in place to help improve the quality of the data collected. For example, the clinician instructs the patient to hold the phone an arm's length away from their mouth, so the phone is parallel to their face. Furthermore, they are instructed to position the phone with the screen facing them and level with their mouth. The SpiroSmart app stores the raw audio and other patient metadata into a SQL database, utilizing the phone's internet to upload to the SpiroSmart server.

The patient is given some time to rest between efforts to avoid breathing fatigue. Once a session is completed, the clinician makes sure the SpiroSmart data is uploaded to the SpiroSmart SQL database and the EasyOne ground truth data is uploaded to a separate Spirometry 360 FRS database. The study protocol calls for 5 ground truth EasyOne trials and at least 3 SpiroSmart audio trials, although this is heavily dependent on the patients' health and cooperation. As a result, there is an inconstant number of trials in differing sessions and therefore, not always a one to one mapping between EasyOne ground truth and SpiroSmart audio. Additionally, there is variation among different clinics as their respective data collection procedures differ immensely and do not always follow the recommended protocol. All of this results in a complex chaotic dataset in desperate need of some organization and cleansing.

11.2 Interpretation

In order to sufficiently outline the cleansing of the dataset, a foundation of what the dataset is comprised of and how it is structured is necessary. First, recall a session is a unique set of trials for a given patient and within a trial, ground truth data, as well as experimental audio data, is collected. Sessions occur one at a time at a given clinic, but different clinics may run different sessions in parallel. Finally, each clinic has a set of smartphones for collecting data. The following unique identifiers are introduced to keep track of all of these concepts; *patient id* (PID), *clinic id* (CID), *spirometer id* and *device id*. Unfortunately, the PID is not guaranteed to be unique because different clinics may create the same PID for different patients. All spirometer and device ids are fixed before data is collected to avoid similar conflicts. To keep track of sessions for a given PID, *session id* (SID) is used and for ground truth within a session, *trial id* (TID) counters are used. Similarly, for the audio data in a session, an *audio id* (KID) counter is employed. See Figure 11.1 for an illustration of how the various ids are

applied to the dataset. It is important to understand the nested nature of the ids in order to follow what is to come.

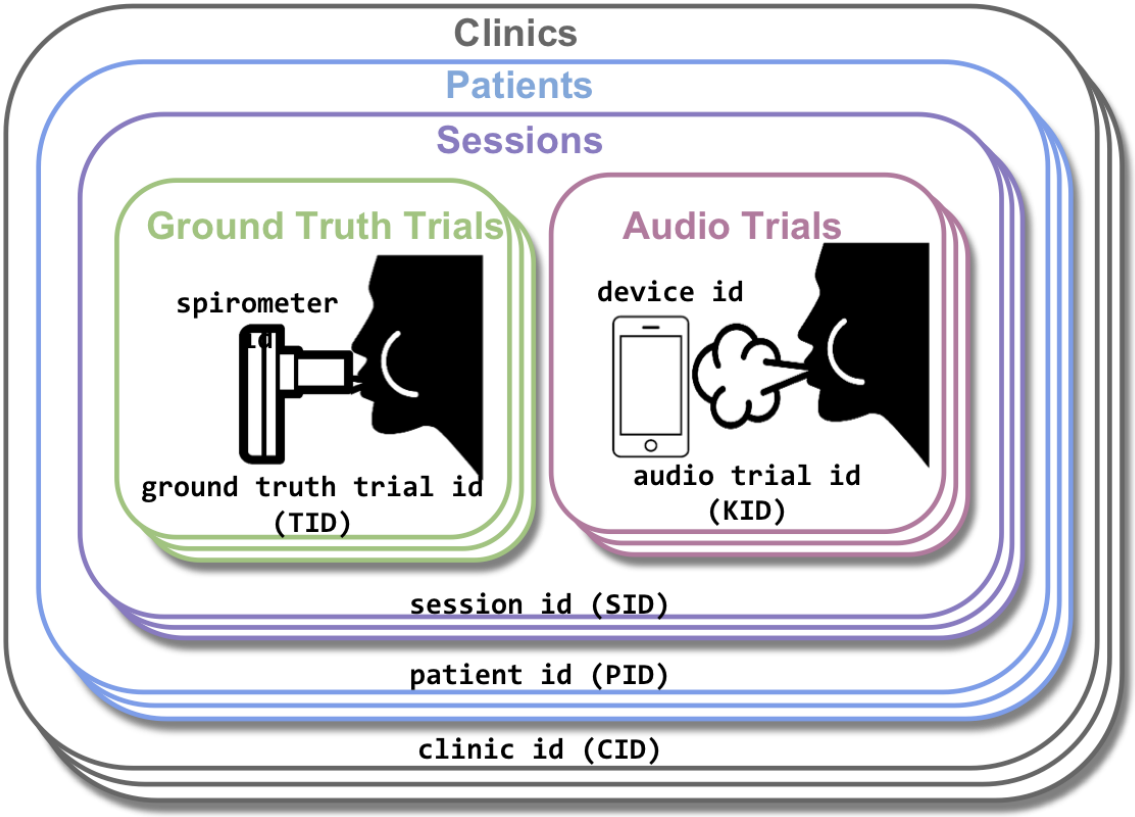


Fig. 11.1: Illustrates the hierarchical nature of the data, every child inherits the ids from it's direct parent so a given audio recording (KID) should link to a session (SID), patient (PID) and clinic (CID)

SQL Tables

Clearly, there is an inherent complexity to the way the data is collected and unsurprisingly this complexity exists in the way the data is stored as well. Several years ago, the pioneering authors of the original SpiroSmart paper deployed a SQL based database schema to store all of the essential data in a way that made sense and fit with the research being conducted at the time. Several distinct tables were created to pair device ids and spirometer ids to clinic ids, match audio trial KIDs to PIDS, and link PIDS to various SIDS and so on. Additionally, there is a large amount of research-based data that is not relevant to the problem space presented in this work. The final schema consists of over ten tables that need to be carefully cross-referenced in a particular way to accurately get the whole picture. It also lacks any form of documentation. In order to migrate away from the archaic structure of the original SpiroSmart database, significant work needed to be done to join all of the tables together into one table complete with all of the information required for developing a new algorithm. Unfortunately, the critical information necessary for unifying all the relevant info into one table is missing, for example, there is no obvious way to match a given KID audio recording

to a SID making it difficult to pair the audio with the ground truth. Since the original authors are for the most part out of the picture, the best way to solve this discrepancy is to turn to unsupervised machine learning. In the next section, a clustering algorithm used to unify all of the SQL tables is presented.

11.3 Clustering

In the initial phase of interpreting the existing data, with the goal of including it all into one table, some critical issues emerged. These issues, highlighted in the previously can be summarized into two key points:

i) *PIDS are only unique within a clinic. This means that a given PID is not guaranteed to map to a single patient in the whole set of all clinics.*

ii) *There is no obvious one to one mapping from audio trials (KID) to ground truth trials (TID).*

Even though audio trials are completed adjacent to ground truth trials, the SIDs only appear in the table with the ground truth TIDs and not in the table with the audio KIDs. It appeared impossible to match KIDs to TIDs for patients who had several sessions since it was unclear which KIDs belonged to which sessions. Furthermore due to issue *(i)* it was uncertain if KIDs for a given PID were actually from the same person. Fortunately, the spirometer ground truth and the audio files have an associated timestamp embedded in them. As long as the ground truth and audio can be grouped for a given PID, timestamps can be used to match ground truth with the nearest recorded audio. This solves the issue *(ii)*, but how can the PID uniqueness issue be solved? It turns out each clinic is more or less in a different country and the timestamp also contains a timezone which can be used to identify the clinic (assuming the clinics have patients at reasonable hours in the day). With this insight along with its reasonable assumptions, all of the conflicts in the database can theoretically be resolved with some form of grouping by PID and clustering by timestamp/timezone.

Before going into the algorithm that solves this, a concrete example will be presented to hopefully clarify the insight given in the previous paragraph. Let's say there are two patients, Alice and Bob. Alice is visiting a clinic in Greece at noon (2 am PST) and Bob is visiting a clinic in Seattle at noon PST. Both of these clinics are brand new and they each eagerly assign their first patient *PID=1*. This is Alice and Bob's first session so they are assigned an *SID=1*. They will each do five trials where each trial has ground truth *TID=[1 to 5]* and audio *KID=[6000 to 6005]*. After Alice and Bob complete their respective session, their data is uploaded to the SQL database. Among other things, there is a table that maps their *PID*

–> *SID* –> *TID* and another table that maps *PID* –> *KID*. Now skip ahead a few months, now each of them has done several sessions at their respective clinic and have multiple sessions. Assuming the KIDs are counted up each time audio is recorded, there is clearly no easy way to match a KID to a TID without looking at the timestamp. Similarly if *PID=1* is queried, both Alice and Bob's results will be returned because the database has no way of knowing Alice and Bob are different. The query would have to be *PID=1 from Seattle* or *PID=1 from Greece*, but first, it must be known which PIDS visited which clinics. Hopefully, it is now more clear why timestamps/timezones can help solve these discrepancies.

11.3.1 Algorithm

In this section an algorithmic approach to unifying the data is outlined:

1. **Combine**: The ten or so SQL tables can be combined into two large, complete tables:
   a) Indexed by TID where each row contains all of the ground truth information
   b) Indexed by KID where each row contains the audio recording and related information
2. **Cluster**: In order to combine the TID and KID table, a mapping from one to another is required. Unfortunately, due to issue *(i)* and *(ii)* such a mapping does not exist as there are collisions and missing information that prevent it from being done accurately, so trials must be grouped into session clusters for both audio trials and ground truth trials. Then, unsupervised clustering is used to exhaustively generate an acceptable mapping by grouping audio sessions with ground truth sessions that match chronologically.
3. **Squash**: The ground truth sessions need to be squashed such that each session is represented by a single best effort. Standard reproducibility rules are applied and if a session is not reproducible, it is omitted from the dataset. Note, each audio session still has multiple audio recordings. So instead of a many to many mapping it is now many to one.
4. **Merge**: Following clustering and squashing, the two tables can finally be merged. For the many KID to one TID mapping, the ground truth is simply copied to each audio row with a small amount of rounding noise added such that each KID from the same session has slightly different ground truth. The final unified table is indexed by KID and each entry has an audio file and respective ground truth.

The clustering processes is explained in detail in the following subsection.

Cluster Ground Truth

Starting with issue *(i)*, conflicting PIDS: the key to matching trials to patients, even when the same PID is used for two different patients, is timezone. If the patient table for *PID=1* has

ten entries consisting of three different timezones, it can be inferred that *PID=1* was used in three separate clinics and using the location embedded in the timezone, the clinic can be identified and recorded in a new CID column. This timezone based clustering effectively solves the issue of conflicting PIDs. The only caveat is to query data for a specific patient, the PID and CID must be known.

Issue *(ii)* is more complex, but can benefit from the new CID cite column as now patients are uniquely identified using the PID and CID. Instead of clustering by timezone since that is now encoded in CID, the timestamps can instead be used to chronologically sort the trials for a given patient. Timestamps are then clustered by date and trials occurring on the same day are considered a session. This may seem redundant since the ground truth has a SID column, but it was found that many entries were missing this as some clinics omitted or were unaware of this metric. With these day clusters, each ground truth entry can be assigned a new SID for each date containing trials and a TID from 1 to a number of trials in that session. Now the ground truth table is complete, trials are sorted chronologically, bundled into sessions and mapped to correct patients. All that remains is clustering the audio table into sessions, then mapping the audio sessions to the ground truth sessions.

Cluster Audio

A similar strategy is employed to apply the concept of sessions to the audio trials. First, group the KID indexed audio rows by PID, forming patient tables, then use the timestamps/timezones to identify the CID, SID and chronological order of audio trials (KIDs). One challenge specifically with the audio is the timestamp and timezone is not a column in the audio table like in the case of the ground truth table.

Thankfully, the audio file itself has an embedded date created tag, so each file had to be downloaded and analyzed to obtain the timestamp. This timestamp is in Coordinated Universal Time (UTC) so it can be used for chronologically ordering the data, but is not useful for inferring the CID since UTC is the timezone for all audio. This is where unsupervised learning comes into play. The manual way to sort this out would be to convert the UTC time to each timezone for all of the clinics, then see which timezone conversion fits best with the time of the ground truth trials. This is very tedious to do manually as there are several thousand PID collisions each containing several trials that would need to be cross-referenced. For this reason, the following unsupervised clustering approach is employed.

Unsupervised Clustering

The exact steps of the manual method explained above can be used to define a machine learning problem. Essentially when a CID is unknown, all the relevant timezones need to be applied to the audio timestamps and then compared to the ground truth time which already has the timezone applied. The machine learning model needs to choose the timezone that minimizes the time delta or error between the known ground truth trial times and

the unknown audio UTC timestamp. It is presumed the correct timezone will result in a timestamp that is close to the ground truth trial times (i.e., small time delta within 30 min) since audio recordings occur in series with the ground truth samples. Conversely, incorrect timezones will result in a timestamp that is far away from the ground truth time. Upon applying the machine learning strategy, each KID will have a proposed timezone which when applied to the audio's UTC timestamp should minimize the time delta between the audio trial time, t_A and ground truth trial time, t_G. Equation 11.1 below shows the cost function to be minimized, Δt:

$$cost = \Delta t = |t_A - t_G| \tag{11.1}$$

A way of visualizing this problem is shown in Figure A of 11.2. In this figure, the y axis represents the time difference between the unknown timezone KID time and the ground truth time (converted to Pacific time) for each PID along the x axis. There are clearly well defined horizontal trends corresponding to common timezones. For example, a distinct trend around $\Delta t = 14$ hrs is apparent and precisely corresponds to the time change from Pacific to Bangladesh time, which makes sense given the majority of the data came from the Bangladesh clinic. Since the goal of this step is to minimize the time deltas by predicting the ideal timezone, re-plotting Figure 11.2 after applying the predicted timezones is hypothesized to result in a plot with a horizontal trend at $\Delta t = 0$ hrs. Figure B of 11.2 illustrates the hypothesis is valid and the predicted timezones tend to minimize Δt. Since the mean Δt is about 8 minutes with a standard deviation of 20 minutes this issue can be considered solved.

Now the audio data can be clustered by time into sessions and these audio sessions can be matched to the ground truth session closest to it chronologically.

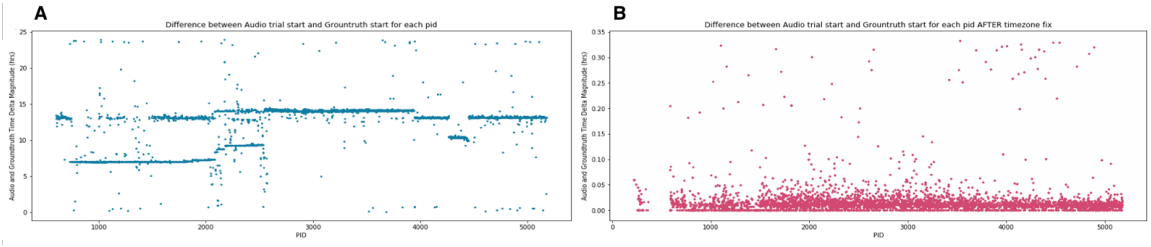


Fig. 11.2: Visualizes the different timezones relative to Pacific Times. In (A) each horizontal trend line corresponds to a timezone shift specific to a clinic, after running the clustering algorithm and estimating the timezone, figure (B) is pl;otted which shows the error between ground truth and audo session times is significantly reduced.

Merge Results

With the audio data and ground truth data aligned timezone wise and grouped into sessions, the two tables are finally speaking the same language and can be merged. To make the mapping one ground truth to many audio, rather than many to many, the best effort is used as the ground truth for each session, assuming it is reproducible (as defined Chapter 5

Spirometry), and the best effort is defined as the session with the maximum FEV1. Some sessions were found to not contain at least three reproducible efforts and were unfortunately omitted, along with the audio.

In total, **5964 ground truth sessions** were identified. Of this, 97.4% demonstrate reproducibility and 98.6% of those have at least one corresponding audio KID. Originally there were 51940 KIDs, and after unifying, 83% were found to have reproducible ground truth resulting in a table with **43318 audio entries**.

This final unified table is constructed so each row entry is a unique audio KID and the columns correspond to ground truth and audio information which map to that KID. This result, table 11.1, has many seemingly duplicate rows as several sequential KIDs share the same ground truth because they are from the same session. To prevent future models from overfitting to these duplicate entries, random Gaussian noise is sprinkled into the ground truth that fits a distribution proportional to the standard deviation of the reproducible entries for each session. This allows the machine learning models to train on a more realistic distribution of ground truth. This table is used for building the models outlined in Chapter 12 Methods, any future mention of the dataset in the context of this research will refer to this unified table.

Tab. 11.1: Final unified table indexed by KID, note that the best effort is used for spirometry ground truth

KID	PID	SID	Trial	Timestamp	Audio File	Spirometry Metrics
1000	1	1	1	3:00	1000.wav	best ground truth effort
1001	1	1	2	3:04	1002.wav	from spirmometer
1002	1	1	3	3:15	1002.wav	and predictive metrics
1006	2	1	1	2:13	1006.wav	
1008	2	1	2	2:20	1008.wav	
1012	4	1	1	9:30	1012.wav	

While all of this clustering may seem meticulous and over-engineered, it is absolutely essential in creating a unified table suitable for building predictive machine learning models, especially given the huge variance in data collection and uploading practices as each clinic. In any data-driven problem, it is imperative that a high degree of confidence exists in the data and each observation (in this case audio) is correctly matched to ground truth. Any error in this critical step will lead to inaccurate predictive models that are difficult to troubleshoot. Alot of this hassle could have been avoidied if the unique SIDs were assigned to each session such that the CID and PID could be obtained given the SID and if the PIDs were assigned to actually be unique. Future data collection efforts will certainly aim to prevent these sort of organizational issues from snowballing as they have in this case.

11.4 Audio Inspection

With the ground truth data sorted out and matched to all KIDs in the SQL database, the next step is to fetch and manually inspect the audio files corresponding to each KID. For this research, binary inspection labels are used and referred to as *keep* and *delete*. When fetching the 43318 KID waveforms, only 26304 resulted in downloadable audio files. The most likely reason is that some audio failed to upload or became corrupted. In addition, there were a number of exact duplicate audio files represented as different KIDs. Since these duplicates are not useful for training, they identified and removed prior to binary labeling. Statistics related to the downloaded audio files are displayed in Table 11.2.

Tab. 11.2: Downloaded audio file statistics

File Count	Unique PIDs	Average Time	Total Time	Format	Sample Rate
26304	6077	7.65s	56 hrs	.wav	32kHz

This task is tedious but simple, one simply must inspect each file and label it as *keep*, meaning it will be included in the training dataset, or *delete*, meaning it will be omitted as it doesn't sound like a valid effort. To speed up this task, waveform plots of each audio file were rendered and displayed in a grid view. Criteria for delete include no obvious exhale effort in audio, multiple efforts, loud background noise or speech, short or incomplete effort and coughing in the effort. An example of a keep and delete waveform is illustrated in Figure 11.3.

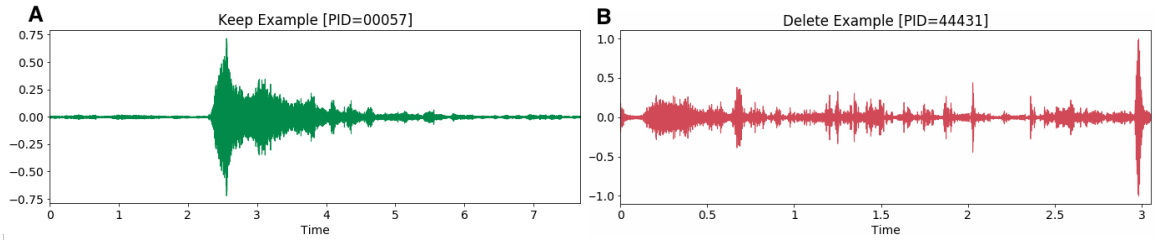


Fig. 11.3: Example spirometry audio recordings. (A) shows a *keep* example with a clear exhale effort and (B) shows a *delete* example with speech and no clear exhale

After screening all of the audio, **20505 were labeled *keep* and 4633 as *delete*** the remaining 1166 were removed as they were marked as duplicates. The files labeled as *delete* are still stored as they are used for the quality control confidence model outlined in the Methods chapter, but they are not used to train the predictive spirometry model. Since the unified table created in 11.3 still has 43318 KID entries, it is pruned down to 25138=20505+4633 entries with labeled audio. Furthermore, the keep/delete labels are added to the table as an additional column.

11.5 Distribution

Now that all of the audio and ground truth data is sorted out, data statistics and visualizations can be generated to illustrate the story told by the data.

Patient Demographic

The Figures in 11.4 convey the statistics for the patients included in the dataset. Upon inspecting the figures, a clear bias toward the Asian demographic is revealed. Since the majority of the data was collected in the Bangladesh clinic and other clinics in Asia, it is not surprising that the racial demographic is dominated by the Asian population. While this is concerning and must be addressed in future work, previous literature indicates the predictive power of race relative to lung health is fairly minimal relative to the other far more important metrics (height and age) [20]. Based on the statistics, the average patient is a 5 ft 3 inch, 148 lb, 38-year-old Asian male, although several children and elderly patients are also present in the data. Future data collection efforts will strive to further diversify the patient demographic statistics as it is critical for the dataset to model the global distribution in order to confidently claim the predictive models are not biased towards certain demographical properties.

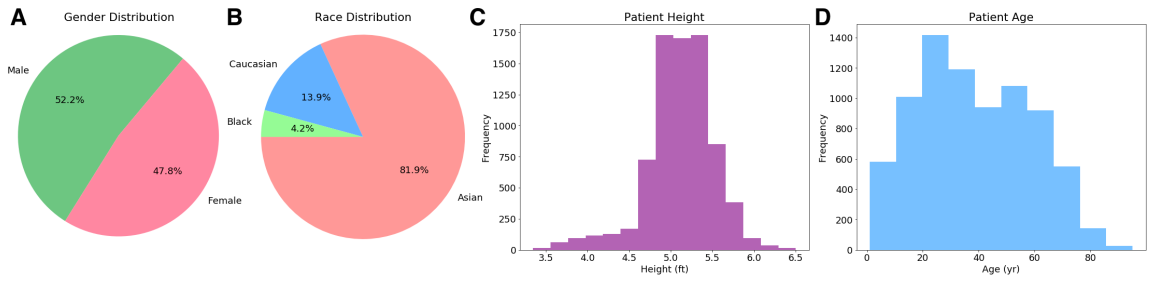


Fig. 11.4: Dataset distribution for (A) race, (B) gender, (C) height and (D) weight

Patient Health

The rest of this chapter assumes a more advanced understanding of spirometry and the various associated metrics which are covered in the Spirometry chapter.

The patient demographics certainly are important as they are used to infer the predicted lung health of the patients. Figure 11.5 shows a histogram of the FEV1/FVC ratio, FEV1 percent of predicted and FCV percent of predicted. Each histogram shows the healthy cutoff as defined by the ATS criteria. Clearly, a number of PIDs are unhealthy or right on the border for FEV1 and FVC percent predicted. The FEV1/FVC ratio metric seems less effected by the imbalance, most likely because many patients have a low FEV1 and FVC, which is less obvious when only the ration is observed.

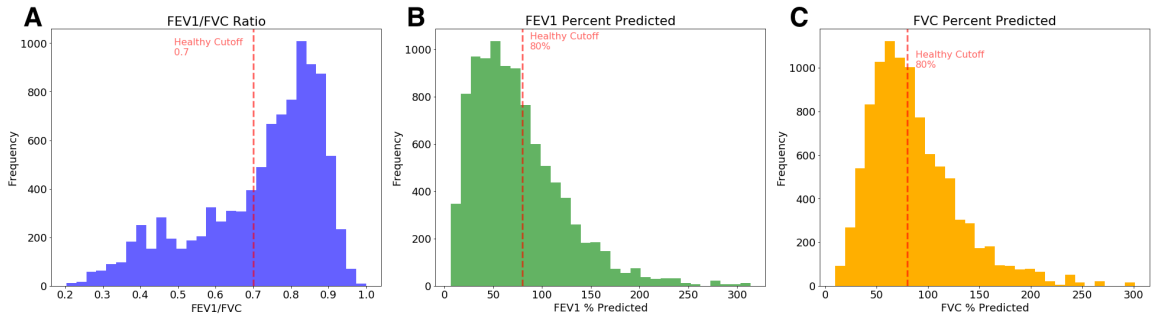


Fig. 11.5: Shows (A) FEV1/FVC ratio with healthy cutoff ratio at 0.7, (B) FEV1 percent of predicted and (C) FVC percent of predicted both with the healthy cutoff at 80%

Using the decision tree shown in Figure 5.4, it is possible to estimate the diagnosis of each entry in the dataset based on the information in the histograms in Figure 11.5. The pie chart in Figure A of 11.6 shows these results of such an exercise, although to truly diagnose the patients, more tests would generally be required. About a third of the patients are healthy, meaning that their FEV1/FVC ratio, FEV1 and FVC percent of predicted are above the cutoff. The rest have some from of a restrictive or obstructive pulmonary disorder. While the dominance of unhealthy people may seem like a blessing given how rare these patients are globally, it also serves as a huge limitation for any data-driven predictive model. If a model is trained on a distribution skewed towards unhealthy people, it can not be expected to perform as well on healthy people. This poses a problem since the global population is much more skewed toward healthy rather than unhealthy.

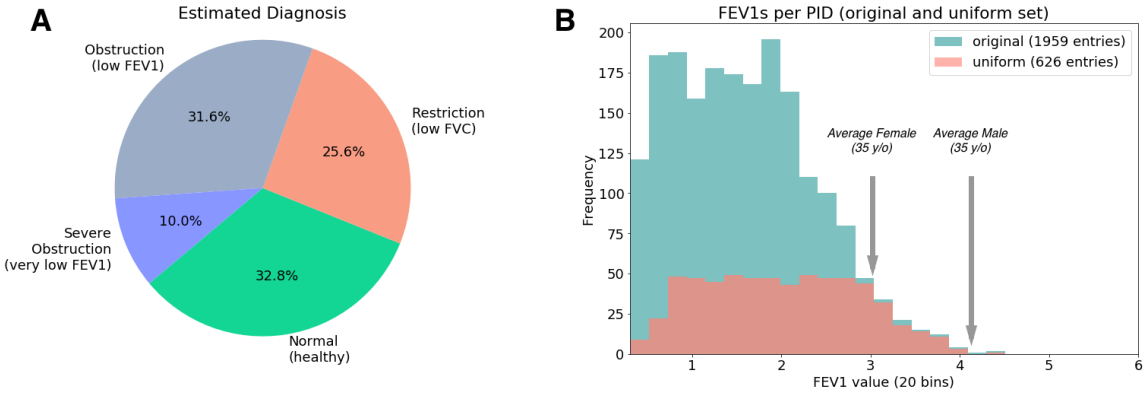


Fig. 11.6: (A) Shows the estimated diagnosis the patients, (B) shows the FEV1s of the original and uniform dataset, as well as the average FEV1 for an average 35 year old male and female

Data-driven models will bias towards the dense areas in the data distribution, which is not an issue when the training data serves as an accurate sample of the population but can be disastrous if the real world severely differs from the training data. Unfortunately, in this dataset, the latter is the case. If the FEV1 average per patient is plotted in a histogram as shown in the blue bars in Figure11.6, it is clear there is a nonuniform distribution of FEV1s,

and it is particularly concentrated at the low < 2 FEV1 values. For reference, the figure is annotated with the FEV1 for an average 35-year-old average sized male and female.

This issue is the single largest limitation preventing the research presented from working to its fullest potential. One way to mitigate the imbalance is to manually create a more evenly distributed dataset by only including patients so the FEV1 distribution is uniform. This way of sampling is shown as pink bars in Figure 11.6. This strategy will allow the machine learning models to learn a less biased prediction model but at the cost of losing the majority of the data. The forced uniform sampling has approximately one third the amount of entries and still is missing the > 4 FEV1 entries. The only true solution is to collect more data in a more thoughtful way so the final dataset serves as a proxy for the global trends. Efforts to collect more uniform data are in progress and covered in the Conclusion chapter.

11.6 Spirometry Ground Truth

While high-level metrics like patient health and demographics are important to investigate, a great deal can also be learned from observing the statistics of the ground truth metrics recorded by the spirometer for each patient. Table 11.3 summarizes these statistics for the main spirometry metrics of interest, namely, FEV1 Predicted, FEV1, FET, FVC, and PEF. Figure 11.7 overlays histograms for each metric which illustrates the mean and variance of each metric. It can be inferred that the metrics with the larger variance will be more difficult to predict as they are less consistent across the dataset.

Tab. 11.3: Statistics for key ground truth entries

	FEV1 Predicted	FEV1	FET	FVC	PEF
Mean	2.54	1.83	7.42	2.46	4.66
STD	1.34	0.93	2.79	1.01	2.00
Min	0	0.26	1.63	0.38	0.76
25%	2.12	1.12	6.10	1.72	3.1925
50%	2.55	1.75	7.20	2.33	4.555
75%	3.01	2.45	8.4	3.0575	5.88
Max	39.94	6.02	27.58	6.75	11.49

11.7 Conclusion

This chapter covers the generation, cleansing, unification, and statistics of the spirometry dataset which turned out to be quite the perplexing task. The following Methods chapter will propose a collection of features and machine learning models that will train and evaluate on the data outlined in this chapter. Given the limitations in the dataset, the goal is not necessarily to create a universal spirometer replacement as much as it is to evaluate

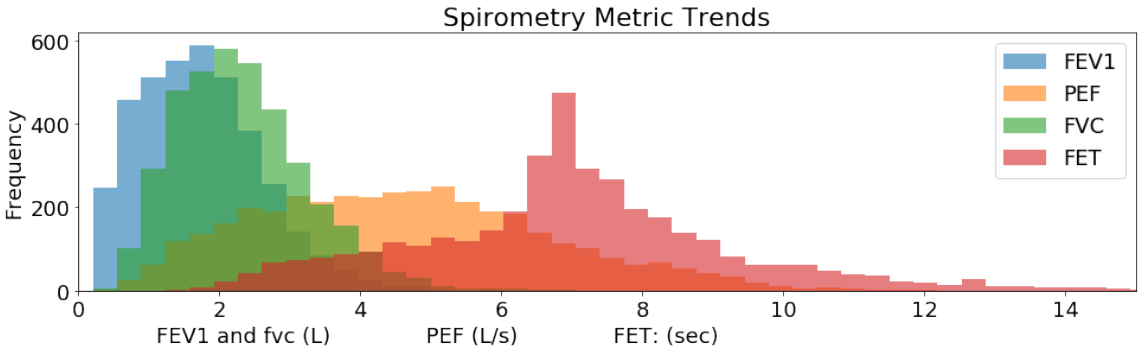


Fig. 11.7: Visualizes the spread of the key Spirometry metrics for the PIDs in the dataset. The different x axis are overlaied and labeled with their respective unit.

how different machine learning techniques perform at the task of measuring airflow from the sound of an exhale. Regardless, the steps outlined in this chapter have resulted in a much higher quality, easier to work with dataset that is hopefully less frightening to future researchers. Additionally, important lessons were learned which will hopefully prevent future data collection efforts from making the same mistakes.

12 Methods

In this chapter machine learning models will be proposed as potential solutions to sound-based spirometry assessment. The methods combine clinical guidelines outlined in the Spirometry chapter with machine learning strategies explained in the Machine Learning Background chapter. First, recall the problem statement:

Explore data-driven methods for computing spirometry metrics suitable for respiratory disease management and screening from a smartphone sound recording of a forced expiratory maneuver.

The exploratory nature of the problem statement implies multiple methods will be evaluated. This chapter will cover the most promising of attempted methods, as well as the strategy used to evaluate them. All of the proposed solutions are heavily reliant on machine learning, although it will be shown there are aspects derived from the physical nature of airflow and the human respiratory system.

Pipelines

The proposed methods are all based off a few abstract pipelines which are created for this research to serve as a reusable codebase for experimenting with machine learning models using the large dataset covered in the Dataset chapter. The pipelines can be categorized as followed:

- **Preprocessing**: includes operations, such as trimming which prepare and standardize the audio data uploaded from the smartphone for feature extraction.
- **Classical Machine Learning**: Implements a fully-featured training framework which supports many of the models covered in Section 8.2, and automatically extracts manually defined features from the preprocessed audio.
- **Neural Network Learning**: Implements a powerful deep neural network training framework with spectrogram input, that supports the architectures covered in Section 8.3 and automatically extracts configurable spectrograms from the preprocessed audio.
- **Evaluation Pipeline**: Supports evaluating any of the models created in the classical or neural network training pipeline on a specified evaluation set.

Spirometry Models

The pipelines were built generically to support other sound or image-based machine learning problems, but they are used specifically in this research to build the following spirometry specific models:

- **Trimming**: Identifies the start and end time of a spirometry forced exhale in a given audio file, then trims the audio so only the exhale region remains.
- **Confidence**: A basic quality assurance system for assessing if trimmed audio contains a valid spirometry effort. The confidence model assigns a confidence score to the audio and if it does not meet the quality standard, the audio is rejected.
- **Prediction**: Estimates spirometry metrics for audio that passes the Confidence model's acceptance criteria. Two Prediction variants are implemented:
  - **Scalar** : Only predicts scalar spirometry metrics such as FEV1 or FVC
  - **Curve**: Estimates a flow versus time curve and uses it to derive other spirometry curves such as FV and scalar metrics.

Sprio AI System

The models can be combined to create an end-to-end system which takes audio directly recorded from the phone and outputs spirometry metrics if the audio passes the quality assurance phase. The proposed system, known as *Spiro AI* is shown in Figure 12.1.

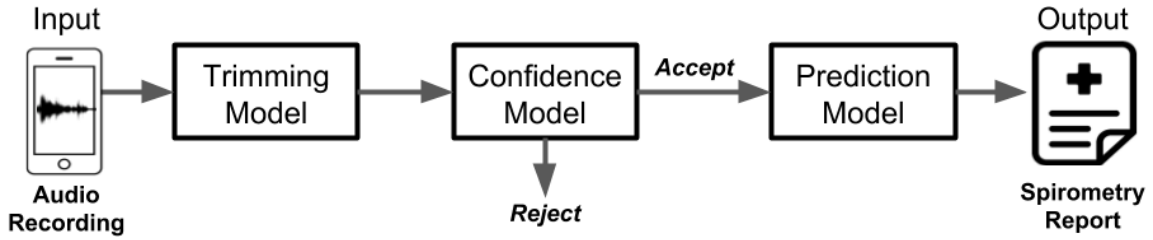


Fig. 12.1: A block diagram of *Spiro AI*, the proposed end-to-end sound based spirometry system comprised of various models

The remainder of this chapter will go into greater detail on the pipelines as well as the specific models that utilizes the pipelines for training and evaluation.

12.1 Preprocessing Pipeline

The preprocessing pipeline has the job of standardizing audio that may come from different phones or clinics which may record the audio in a slightly different format. For example, some clinics start the recording up to a minute before the patient actually performs the

maneuver, while others start it right before the exhale. Other sound specific parameters such as sample rate number of channels and bit depth may vary as well. With a dataset as large and variable as the spirometry one used in this work, it is imperative that all of the sounds are converted to a familiar format before any model training can be performed effectively. The preprocessing pipeline can be represented as a series of steps outlined below:

1. Load and decode the audio file into an array of integers
2. Convert the array to be a single channel, mono array
3. Resample the array to have a sample rate of 16000 Hz
4. Apply optional sound processing operations such as normalization or filtering to the audio array
5. Apply the Trimming model to trim the audio vector so the trimmed result starts just before the exhale and ends just after the airflow terminates
6. Output the trimmed, standardized audio sequence for further processing

12.2 Classical Machine Learning Pipeline

The classical machine learning pipeline, or ML pipeline for short, takes a collection of preprocessed audio, extracts manually defined features, then exhaustively trains a set of machine learning models using k-fold cross-validation to prevent overfitting and grid search to choose the optimal model parameters. It is implemented in Python and relies on the open source scikit-learn python library for the ML models implementation with exception of gradient boosting which utilizes the superior LightGBM library. To use the ML pipeline, the input dataset must be specified, as well as the ground truth label that the trained model must learn to predict. The ML pipeline only works with single-valued (scalar) outputs for both classification and regression problems. Due to the single output limitation, the ML pipeline cannot train models to predict spirometry curves or multiple metrics. Instead, an independent model must be trained for each desired scalar metric. Nonetheless, the ML pipeline can be used to build a confidence accept/reject classification model, as well as a series of models for the spirometry metrics of choice. Because this pipeline leverages the classical machine learning methods, manually defined features must be extracted ahead of time. This section will start by describing the manual features used in this work.

12.2.1 Manual Feature Extraction

A universal set of features is defined for use in both the confidence and predictive models. This decision was made for convenience and efficiency reasons. Given the massive scale of the training data, having different sets of features for different models overly complicates the

process and makes evaluation more convoluted. Furthermore, assuming these models are deployed, it is much more efficient to extract one set of features for both the confidence and prediction model, rather than require separate, computationally heavy feature extraction for each model.

Many of the extracted features are alluded to in the Sound Background chapter and will not be explicitly defined in this section. The features must be in the form of a table where each column represents a feature and each row represents a different entry in the training set, indexed by audio file KID. Some features are single, scalar values, while others are an array list of numbers. Any subset of these features can be used to train a classical ML model, and it will be shown that different variations are compared performance wise in the Results chapter. The universal feature set has a total size of 140 columns and is summarized below:

Scalar Features

- **Loudness**: The average loudness in dB of four key segments in the waveform, namely: 1) the whole audio segment (total), 2) the first 200 ms (explosive region), 3) everything after the first 200ms (the decay region) and 4) the last 100ms (room noise).
- **Filtered Loudness**: The loudness of the same four regions defined above, except the waveform is low pass filtered to attenuate frequencies above 1kHz. This effectively removes sound outside the bands where airflow sound is expected to exist (the low frequencies).
- **Area**: Simply the sum, or area of the waveform integrated with respect to time. Also includes an additional variant which computes the area of the low pass filtered waveform from above.
- **Duration**: The total time in milliseconds of the waveform (note, the audio has already been trimmed by the Trimming model, so duration reflects how long the exhale lasted)
- **Peak Count**: Designed with the confidence model in mind, this feature quantifies the number of distinct peaks using a standard peak counting algorithm. The intuition is there should only be one peak when at the time when peak flow occurred. Additional peaks are an indicator of a low-quality effort which may have been trimmed incorrectly or contain a cough or multiple efforts.

Array Features

These features must be expanded such that each array index has its own column in the features table. Many of these are shown in figures in the Sound Background chapter.

- **Amplitude Envelope**: An amplitude envelope of the audio downsampled to 16 samples. A low pass filtered variant is also included for a total of 32 points.
- **Polynomial Coefficients**: The uncompressed amplitude envelope is fit by an 8th order polynomial to smoothly trace the variation in amplitude. Only the last four coefficients

are used as the first few coefficients were found to be largely the same no matter what the input.

- **Spectral Envelope**: A spectral envelope conveying the power spectral density, downsampled to 16 samples.
- **Mel-Scaled Spectrogram**: A typical Mel-spectrograms is downsampled and compressed such that 80 values summarize the information in the original 128x64 spectrogram.

Prior to training these 140 features are extracted and arranged in a tabular form. This step is usually performed once and then the features can be reused assuming the training data does not change. The feature extraction process takes a few hours, but the result is a table which summarizes several gigabytes of audio in less than a few megabytes. Some of the features may be redundant, but part of the training objective is to assess which features are important for a given model. Next, the training structure will be outlined for both the classification and regression cases.

12.2.2 Supported Models

The following section lists the supported models as well as the procedure for training them.

Binary Classification

Recall binary classification tasks usually output a probability of an event being true, between 0 and 1, then the output is binarized based on whether or not it exceeds the decision boundary. During training, all classification models use accuracy as the metric to maximize, although this can be changed. There are several potential models that can be utilized for binary classification, but for this work, the list is limited to the following models, which are ordered in terms of complexity with their abbreviation in parenthesis:

- Naive Bayes (NB)
- Logistic Regression with L1 Regularization (Log L1)
- Logistic Regression with L2 Regularization (Log L2)
- K-Nearest Neighbors (KNN)
- Random Forests (RF)
- Gradient Boosting (GBM)

Regression

Regression models must output a continuous number rather than a binary decision and therefore represents a class of models separate from classification. As it turns out, many of the classification models can be re-branded for regression, so the list ends up begin similar. The models are configured to minimize mean squared error between the prediction

and ground truth value. The following models are used for regression, also in order of complexity:

- Mean (guessing the mean value every time)
- Linear Regression with L1 Regularization (Lin L1)
- Linear Regression with L2 Regularization (Lin L2)
- K-Nearest Neighbors (KNN)
- Random Forests (RF)
- Gradient Boosting (GBM)

Training

There are a few other common practices used to train models using this pipeline. First, k-fold cross validation is used. Cross-validation is used to retrain the model multiple (k) times on different subsets of the training data to ensure the trained models are not overfitting to the training set. In this work, a k value of 5 is used. Many of the models, such as KNN, RF, and GBM require additional parameters to control the depth or complexity of the model. It is often difficult to manually tune these parameters, so a strategy known as grid search is utilized to converge to the best parameter choice by retraining the model several times, each time with a different value for the parameter being tuned. The parameter from the best performing model found in the grid search is used for this work. Upon completion of training optimized model is saved along with other information including the training error, elapsed time, feature importance and grid search optimal parameters. The size of the outputted model depends on the type of model trained and ranges between a few kilobytes and less than 50 megabytes. The elapsed time is anywhere between 1 and 30 minutes depending on the machine used, input data and how cross-validation and grid search are parameterized.

Conclusion

In summary, a classical machine learning model for classification or regression can be trained on a set of manually configured features using this pipeline. These models can only be trained to output a single value. The resulting models are trained with 5-fold cross-validation and parameters are optimized using grid search.

12.3 Neural Network Pipeline

The neural network training pipeline, or NN pipeline for short, takes a collection of preprocessed audio, generates a configurable spectrogram input and trains a customizable neural network. It is implemented in Python and relies on Tensorflow to generate and train NN models. In order to use the pipeline, the input dataset must be specified, as well as the ground truth labels the trained model must learn to predict. The NN pipeline s very flexible

as an input or output of any dimension can be declared. This enables sequence outputs like curves or multiple spirometry outputs to be specified in a single output. Therefore, all proposed models can be built as a NN using this pipeline, with any input such as the raw waveform, a spectrogram or the manual features. While raw time series audio and manual features are experimented with for this work, the results only cover models trained with Mel-scaled spectrogram input, as they performed best. Furthermore, for this work two main architectures are explored, the CNN and CRNN, which are covered below.

12.3.1 Spectrogram Generation

The NN models in this work used Mel-spectrogram (Mel-specs) with 128 time frames and 64 Mel-bands and can be thought of as a volume of shape *(128, 64, 1)*. To generate the Mel-specs, first, a filterbank with Mel-weights is initialized. Next, the input trimmed audio waveform is set to a fixed length of 5 seconds. If the input is longer than 5 seconds it is trimmed and if it is less than 5 seconds it is padded with 0s. This way all of the Mel-specs have the same timescale. The frequency scale is already fixed since all files share the same sample rate (16kHz) and are thus bandlimited at 8kHz. The fixed length audio is converted to a log-spectrogram using the standard STFT transform (1024 windows with an overlap of 400), then the Mel-filter bank is applied to scale the frequency axis. The result is a 128 frames x 64 bands log scaled Mel-spectrogram in dB. An example is shown in Figure 12.2.

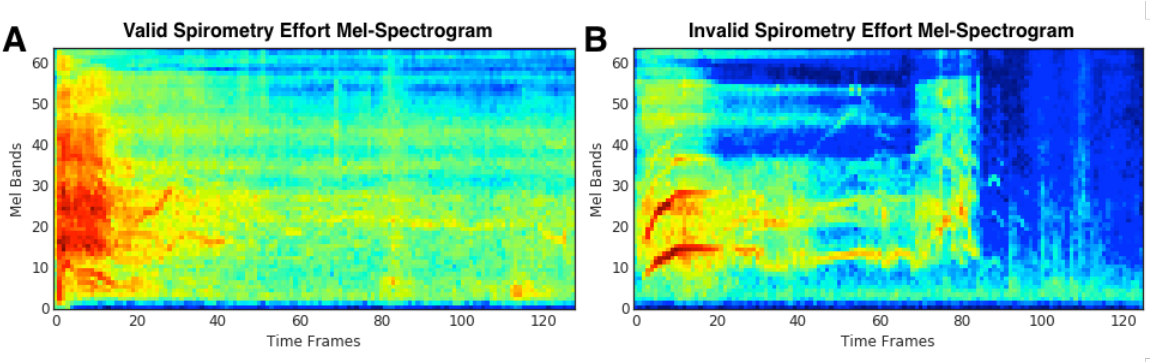


Fig. 12.2: An example of the spectrograms used in the neural network pipeline. (A) shows a valid spirometry exhale, while (B) Is an invalid effort example

A novel feature of this pipeline is its ability to set the weights used in the filterbank as a trainable parameter, then extract the ideal filterbank, post-training. This means rather than using Mel-scaling, the network can adjust the scaling to make the bands of interest high resolution, while the bands with little information are rendered at a low resolution. Mel-scale is an example of weights designed to scale the frequency axis similar to human ears, which is ideal for most natural sounds, but when the scope of sound is narrowed, such as in the case with spirometry sounds, there are certainly more optimal filter-banks. This option allows the NN to handle finding the optimal filter bank. Another useful feature in

the spectrogram generation options is to set a minimum and maximum frequency if the 0 to 8kHz range is undesirable.

12.3.2 Convolutional Net Architecture

The convolutional neural network (CNN) architecture is an n layered CNN with m fully connected layers (FCN) leading to the output. The architecture is defined by connecting a series of 3D blocks of configurable volume. The type of blocks used in the CNN architecture, which are mostly covered in Section 8.3, are listed below:

Blocks

- **Waveform**: A block of volume *(n samples, 1, 1)* which is meant to represent a single channel audio waveform. The only parameter is the *n samples*
- **Spectrogram**: A block of volume *(n frames, n mels, 1)* which converts a 1D waveform block into a 2D spectrogram. The filterbank is either fixed or trainable and has spectrogram generation parameters such as: *n frames*, *n mels*, *min freq* and *max freq*.
- **Convolution** (conv): A block volume of *(height, width, depth)*. Implements a convolution layer with max pooling and a configurable activation function. The *height*, *width* are fixed based on the input to the block and the output *depth* is a definable parameter. All of the other standard convolutional layer parameters are available. By default, the block uses batch normalization, ELU for the *activation*, a *filter size* of 7x7, a max *pool size* of 2x2 and *dropout* of 0.3, but these can all be specified.
- **Fully Connected** (FC): A block of volume (*n nodes*, 1, 1) which serves as a standard FC layer with a customizable *activation*, which defaults to ReLU and *dropout* with a default of 0.3.
- **Output**: A block of volume (*n outputs*, 1, 1) which is a standard FC layer except with a linear *activation* in a regression problem, or sigmoid if classification. The *n outputs* is flexible and can support anything from a sequence to a single binary output

With these blocks, several architecture variants can be assembled. In addition, the blocks can be tweaked or extended as needed. For example to the Convolution, the block has been extended to match the spec for a Gated-Convolution layer which has proven as useful in audio-based NNs [93]. As a result of a lot of experiments and related works, the following architectures tend to offer top performance in both the prediction and confidence models.

Space Needle Architecture

The Space Needle gets its name from the 3D shape it represents when assembled. It is fairly simple to define as it only requires a few parameters which dictate the scale of the architecture. First, the input and output must be specified. The input is typically a Waveform followed by a Spectrogram block and following the configuration specified in Spectrogram

Generation, would have a volume of (128, 64, 1). The output is simply an output block matching the desired output shape. Following the input is a pyramid-shaped series of Convolution blocks where each successive block has a *depth* of 0.5 that of the preceding block. The parameter *n conv* dictates how many blocks will be in the series, note, it is limited by the size of the input block, otherwise, too many decaying convolution blocks would yield a negative volume. The *conv decay* factor, 0.5 by default can also be tweaked (it controls the amount of max pooling) and the other Convolution block parameters can be defined as needed. After the convolution block series is a similar pyramid shaped series of Fully Connected blocks leading to the output block. The FC blocks are flattened, i.e., have shape (*n nodes*, 1, 1), where *n nodes*= the volume of the last conv block. The FC series has the same parameters as the conv series, namely: *n fc* and *fc decay* (which defaults to 0.5). Assembling the space needle with default decay and *n fc* = 4, *n conv* = 3, depth=32 and spectrogram input, and binary output creates a model is shown in Figure 12.3.

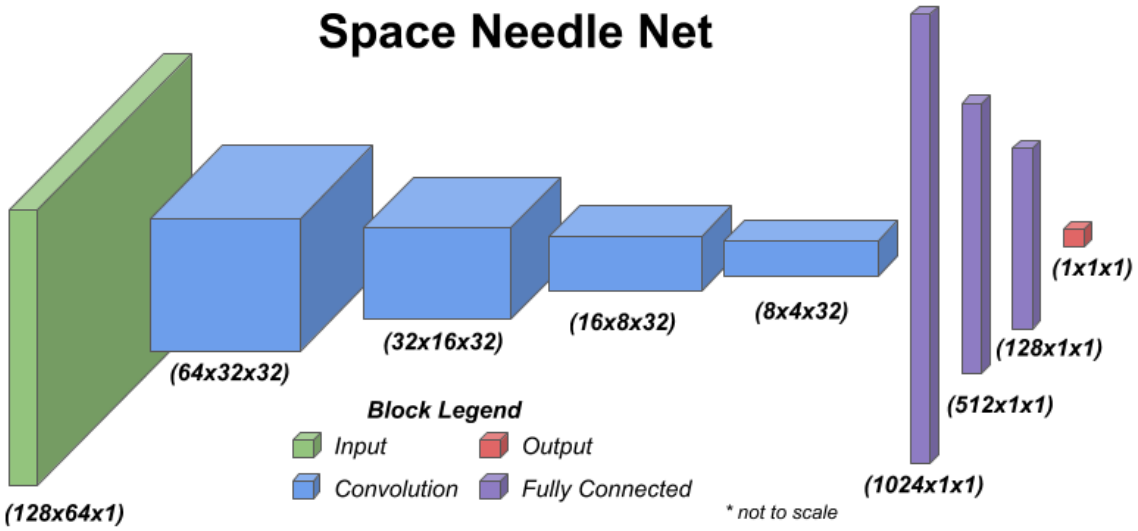


Fig. 12.3: Illustrates the Space Needle architecture with the decaying convolution and fully connected blocks.

CNN Architecture Variants

There are many variants to the CNN architecture proposed above, most of which involve replacing the FC pyramid blocks with a different set of blocks. The convolution blocks perform the job of automatic feature extraction, so they are best left as is, aside from scaling the volumes. Formidable FC block replacements include nothing at all, a global max or average pooling block or a recurrent neural net (RNN). The argument for nothing at all is enough conv blocks can essentially flatten the volume into a near *(N, 1, 1)* shaped block just like the FC blocks. Global pooling essentially forces the last conv block to do exactly this and is much more efficient than a series of FC blocks. Placing an RNN at the end of the network is a common architecture for pattern-based sound known as a CRNN. This effectively gives the final layers the ability to remember historical features extracted by the conv blocks. To

utilize this ability, the input must be fed in in a time distributed manner rather than the full spectrogram. CRNNs are difficult to tune but have been shown to outperform CNNs on the same data (although they train much slower).

Training

The pipeline is typically training using stochastic gradient descent with momentum or Adam as the *optimizer*. In regression problems, mean squared error is used as the *cost* function and binary cross-entropy is used in binary classification problems. Usually, the networks are instructed to train for unlimited iterations, and an early stopping algorithm is used to dynamically terminate training if it shows signs of convergence or begins to diverge, i.e., the cost metric is no longer progressively decreasing. This typically occurs between 100 and 1000 iterations, which can take over 8 hours on a high-performance GPU powered machine. Tensorboard, a dashboard web interface, logs the progress and shows the training/validation cost metric plotted against time so convergence can be monitored. When training concludes, the final model and the trained weights are saved along with the final training set error and elapsed time metrics. The final model is usually between 10 and 100 megabytes depending on how many layers and the volume of each layer. Pre-trained models can also be retrained or used as a starting point for a different dataset.

Conclusion

In conclusion, the neural network pipeline supports the high-level arrangement of blocks representing different neural network components and layers. It is built to encourage sandbox style experimentation and scales well with lots of data or outputs. The integration of Tensorboard allows training models to be monitored and trained models to be easily compared.

12.4 Evaluation Pipeline

The goal of the evaluation pipeline is to be model independent. The two training pipelines described above output several different models, but to the evaluation pipeline, they are merely a "black box" with an input, typically a waveform, or pre-extracted feature and an output of either one or many binary or continuous outputs. The evaluation pipeline must load a model, and perform inference, that is, get the output from the model after supplying the input, on a preset evaluation dataset. It is important that 1) the evaluation dataset used does not contain any overlapping data from the training set and 2) the same evaluation set is used to compare models predicting the same output(s). Therefore, the evaluation pipeline is parameterized by the model, the evaluation set, including the ground truth and the type of output (regression or classification and one or many outputs). Evaluation is much cheaper than training and can, therefore, run on any device with a decent CPU. Upon

completing evaluation which takes a few minutes for most models (depending on the CPU and the evaluation set size), a report is saved with the error metrics and plots as well as well as elapsed time. The evaluation report different depending on whether the problem is classification or regression, so they are described separately. The findings published in the Results chapter are generated using this pipeline.

Binary Classification

For binary classification models, the evaluation report contains the standard classifications such as *precision*, *recall*, *f-score* and *accuracy*. Additionally, the *elapsed evaluation time* is provided. Finally, *precision recall* and *ROC* plots are saved. There is an option to save the outputted predictions as well if further analysis is intended.

Regression

In the regression case, the report contains various error metrics and the *elapsed evaluation time*. The error metrics included for the evaluation set are, *average mean squared error* and *absolute error*, *root mean squared error*, and *average percent error*. When the output is a scaler, the *correlation coefficient* and *Bland-Altman* plot are also recorded. When the output is a curve, the predicted curve overlaid with the ground truth curve are plotted.

Conclusion

The evaluation pipeline offers a simple, standardized way to evaluate any type of models exported by the training pipelines. Different evaluation sets can be used and a report is generated for each evaluated model.

12.5 Spirometry Models

This section covers the proposed models for each block in the system diagram in Figure 12.1. While hundreds of variants are trained over the course of the research, these are the ones being evaluated for this work. See Section 10.3 for details on the training process. The model selection process is mostly based on performance error wise, but also sided for simpler and more efficient variants. For example, the CRNN often performed marginally better, but due to its much longer training time and extra complexity, the CNN version is favored. It is certainly possible in the presence of more or different data, this selection would be different.

12.6 Trimming Model

The goal of the Trimming model is to trim a waveform so only the sound of the forced exhale remains. It is a crucial and underestimated piece in the system. Though its job is simple, almost instantaneous for a human, it is high risk, as any error will propagate through the system and may result in unnecessary rejection by the Confidence model. Trimming is an example of a case where machine learning and certainly deep learning might be overkill. Especially given the unique shape of a spirometry exhale relative to other types of sounds. Therefore a rule-based, old-school AI algorithm is employed as a based case.

Rule Based (unsupervised)

The procedure is quite simple and can be outlined completely as followed:

Note: $\bar{L}$ signifies an average loudness, normally measured in dB, but treated as an log scaled intensity percent where 100% is the loudest. This is more intuitive than dealing with the negative dB scale. Also, many variables like windows size and thresholds can be set to anything, the following values worked well in the manually inspected cases.

1. **Normalize the audio waveform**: Since this is a loudness based algorithm, normalizing the audio such that the max amplitude is the same in all waveforms makes the parameters easier to tune.
2. **Measure the average room loudness**: Which is expressed as the quietest region at the start or end,$\bar{L}_{room} = min(\bar{L}_{[0\ to\ 100ms]}, \bar{L}_{[end-100\ to\ end\ ms]})$. This naively assumes the the first and/or last 100ms only contain the background noise, which is not very robust.
3. **Forward search for an impulse**: This may be the trigger for the exhale event. This search uses a small sliding window of size 5ms and looks for a segment at index n where: $\bar{L}_{[n\ to\ n+5\ ms]} > 2\bar{L}_{room}$, in other words it stops searching when it reaches a segment significantly louder than room noise.
4. **Backtrack to just before the impulse**: Since the goal isn't to start at the peak, but slightly before. This search backtracks with a large window size of 30ms and marks the *start* trim point as the index, n, where: $\bar{L}_{[n\ to\ n+30\ ms]} < 1.5\bar{L}_{room}$, or the point before the impulse that is much closer to the measured room noise.
5. **Forward search for end of decay**: In order to find the *end* trim point, it is assumed the exhale lasts at least 1500 ms. A large window size of 30ms starts at *start* + 1500 ms and forward searches for the index, n where: $\bar{L}_{[n\ to\ n+30\ ms]} < \bar{L}_{room}$, which indicates the noise level has come back down to the room noise.
6. **Continue forward search until another impulse is detected**: Often times the tail end of an exhale is very quiet, rather than prematurely cutting it off, this final search searches for the index, n where: $\bar{L}_{[n\ to\ n+30\ ms]} > 1.5\bar{L}_{room}$, signifying the noise level

has come up again which may mean the patient coughed or is talking. This point is then marked as *end*, otherwise the end is set to the end of the original waveform.

This procedure is trivial to program and works surprisingly well. While no machine learning is used to tune it, human effort certainly helps. About 2000 trimmed recordings are manually inspected and the windows sizes and thresholds are set to prevent the most common issues. Still, this algorithm is purely amplitude based and does not consider the frequency bands of the sounds. Therefore it is prone to false positives as it is easy to generate a completely different sound with a similar amplitude envelope. Examples of successful and failed trimming are shown in the Results chapter.

Cross Correlation (semi-supervised)

A much more elegant way to locate the exhale region without needing to label all of the data involves the concept of cross-correlation. Essentially, cross-correlation slides a *reference* signal shape along the x axis of another, usually longer *source* signal and identifies the x position where the *reference* signal best matches, or correlates to the *source*. It is very similar to the convolution operation.

The way cross-correlation can be applied to trimming requires a *reference* signal to be generated by averaging the amplitude envelope of several already correctly trimmed recordings. This ground truth is assembled by manually inspecting outputs of the rule-based method. Then the *reference* can be cross-correlated with an untrimmed *source* and the maximum point of similarity can be taken as the *start* time. The generated *reference* signal is displayed in the Results chapter. While this method is promising it has not been sufficiently evaluated or compared to the rule-based method.

Neural Network Pipeline (supervised)

The future plan is to bake the Trimming model into the Confidence model, assuming a NN is used for it. CRNNs have shown to be very effective at identifying the start and end in various audio detection tasks [93]. It is compelling to combine the two models as Trimming can be thought of as a part of quality assurance and audio that cannot be effectively trimmed should be rejected anyway. Early investigation into this method has progressed, but without a large labeled dataset, it is difficult to match the performance of the other methods.

Conclusion

Currently, the tried and true rule-based approach is used to trim the existing dataset and all future data. Impotent future work will involve evaluating other more robust methods.

12.7 Confidence Model

The Confidence model serves as a quality check to ensure only legitimate spirometry efforts are sent to the Prediction model when the system use used in a deployed scenario. Since the Predictive models are not physics based, it is undefined how they will perform if the input significantly differs from the training data, therefore the Confidence model is built to ensure the right kind of input is passed through the system. The goal of this model is to eventually run on the phone, so other than accuracy, inference time is also crucial. For this reason, the NNs are configured to be lightweight at the cost of accuracy. Furthermore, since the model is for quality control, it should be tuned to have a low false positive rate, therefore precision-recall and ROC curves will be shown as results.

Dataset

The dataset used originally comprised of the keep/delete labels gathered from the task described in Section 11.4. There are two problems with using these labels for training, first, it is highly imbalanced (20k keep labels and 5k delete) and second, there are many more example of sounds that should not be accepted which are not captured in this dataset. For these reasons, the training set is augmented with approximately 15k "negatives" which are labeled as delete. These negatives are a random collection of sounds from datasets in other audio domains, for example, speech, urban sounds, and coughs. Using the negatives both balanced out the keep/delete and add diversity to the delete set so the model can learn to reject coughs, speech, and loud background noise. Thanks to the preprocessing pipeline, the audio from various sources all has the same encoding. An example of a recording labeled as keep and delete is shown and Figure 12.2 A and B respectively. The final training set is comprised of 20k entries with 50% keep and delete. The evaluation set is only 200 entries, but they are handpicked to be difficult realistic examples. It was found randomly generating the evaluations set made the task too easy and gave misleading results. The ground truth, Y, is simply the binary keep/delete label where 1 is keep.

12.7.1 Models Evaluated

Given the problem is a binary decision, the classification variants of the training pipelines are used. The NN based Confidence model is referred to as ConfidenceNet.

Classical Models

The classical ML models explored are Logistic regression with L1 and L2, Gradient boosting (GBM), Random forests (RF), K-nearest neighbors (KNN) and Naive Bayes (NB). There are three different variants of manual feature sets, X, used; *all* uses all the features defined in Section 12.2, *mel-only* uses only the 80 Mel-spec based features, and *no-mel* uses all of the

features except for the 80 Mel-spec features. The reason for this is to gain insight on which features are the most useful. Furthermore, the *mel-only* results can be directly compared to the neural network Confidence model, which also uses Mel-specs as an input.

In all feature variations, the general order of the models that achieved highest accuracy remains the same: **GBM > RF > Log L2 > Log L1 > KNN > NB**. For this reason, GBM is selected as the decision tree based model and Log L2 is selected as the linear model, and only these two models are further evaluated. It is also worth mentioning the GBM requires significantly less memory and processing time to perform inference, making much more attractive compared to the bulky, RF. Grid search identified 320 to be the optimal value for the number of estimators parameter. In summary a all, mel-only and no-mel variant of GBM and Log L2 will be evaluated.

ConfidenceNet Models

Two ConfidenceNet models are explored, a lightweight CNN and a lighter CRNN. The both use Mel-specs as the input feature X. They each are capable of similar performance in general, but the CRNN was not reliable as each training session it either performed on par with CNN or significantly worse. For this reason, only the CNN is fully evaluated. The CNN Space Needle architecture is defined as shown in Table 12.1:

Tab. 12.1: Parameters used for the ConfidenceNet CNN model

Parameter	***n conv***	***depth***	***filt size***	***pool size***	***n fc***
Value	3	32	7x7	3x3	2

In total, the model is relatively small with only 143 thousand trainable neurons, occupying only 1.2 megabytes when fully trained. It is independently trained several times to ensure training converges to similar weights each time. Unlike in the case with the CRNN, the CNN had very repeatable results. CNN convergence usually occurred in < 100 iterations, usually around 40, which takes about an hour.

12.8 Prediction Model

The prediction model does all of the heavy lifting. It must generate a full spirometry report, from only a trimmed audio recording of an exhale. There are three types of models proposed as a solution to this difficult problem. Two of the approaches, the classical ML, and the Scalar NN, named ScalarNet, train an independent model for each desired scalar spirometry metric. The metrics models are trained to predict are FEV1, FVC, PEF, and FET. When performing inference, each model must independently process the trimmed audio which can be inefficient. Also, by treating the spirometry metrics as independent, the single output models will not learn directly the relationships between differing spirometry metrics and are

therefore blind to some degree. The benefit is they can focus all of the weights and machine learning power to a single output rather than being distracted by multiple outputs. This strategy is more similar to prior work in machine learning based spirometry.

The third curve based option, named CurveNet, is far more novel as it utilizes a NN trained to predict the flow versus time (FT) sequence, then derive all of the metrics and other curves. It has the added benefit of outputting the FV and VT curves in the spirometry report. Additionally, the outputs are all dependent on the original FT curve and thus make sense physically. Since this model does not predict the metrics directly, it can be harder to control and bound the derived outputs. For this reason a custom, physics-based cost function is developed to penalize the derived scalars as well the curves in a way that can be customized to penalize certain outputs more than others.

The goal of the prediction model is to eventually run on-device, so it must be lightweight and comprised of easy to extract features in the classical case. Given CurveNet is a single model which only requires an input spectrogram and can predict all of the spirometry metrics, it is the most compelling case for an on-device model, assuming it is on par accuracy wise with the other options. The most important aspect of the prediction models is low, unbiased error, to assess this, a Bland-Altman plot is generated for the top performing models.

Dataset

The dataset is limited to only entries labeled as *keep*, which contain curves and all spirometry metrics as ground truth. Additionally, the entries all demonstrate reproducibility. After applying all of this criteria, a training set with 14.5k remained. About 5% of the patients in this set are kept out for evaluation. This set of patients is manually picked to represent a somewhat normal distribution in terms of FEV1 to ensure the evaluation set best represented the population. As explained in the Dataset chapter, the data is heavily skewed toward unhealthy, which is irreversible.

For the single output models, the ground truth, Y, is simply the corresponding metric measured from a clinical spirometer. In the case of CurveNet, the ground truth FEV1, FVC, PEF and FET outputs are arranged into an array. The ground truth FV and VT curves are re-sampled so they have a sample rate of 50Hz, and are set to a fixed length of 500, or 10 seconds using padding or truncation. There is no ground truth for the FT curve, so it is not part of the cost function, only the parameters derived from it. Therefore, the ground truth, Y is a vector of size $500+500+4 = 1004$ values, although the cost function penalizes them differently.

12.8.1 Models Evaluated

The nature of this problem is regression, so the respective classical and NN pipelines are used.

Classical Models

The classical ML models explored for regression are: Linear regression with L1 and L2, gradient boosting (GBM), random forests (RF), K-nearest neighbors (KNN) and Mean tracking (simply guessing the mean every time). Each of these are trained with 5-fold cross-validation and grid search for all four spirometry metrics. Just as in the ML Confidence Model, three different variants of manual feature sets, X, are used; *all*, *mel-only*, and *no-mel*.

Similar to the confidence model, only the top decision tree model and linear model is used in the final result. The model rankings resemble the Confidence model rankings, even though the problem and dataset is vastly different, namely: **GBM > RF > Lin L2 = Lin L1 > KNN > Mean**, were ranked in terms of smallest absolute error. In these models, Lin L2 and Lin L1 are pretty much equal, but L2 is used simply to stay consistent with the Confidence model. GBM is once again selected as the superior decision tree model.

ScalarNet Model

A ScalarNet model is used for each independent output, with Mel-specs as the input feature X. The Mel-specs are generated using a Specrogram block with the filterbank weights and frequency cutoffs set as trainable parameters. The final results will show what the ideal spectrogram scaling turned out to be. There was exploration into the architecture parameters, but ScalarNet by far had the least experimentation. It also uses the CNN Space Needle architecture and is parameterized as shown in Table 12.2:

Tab. 12.2: Parameters used for the ScalarNet CNN model, each independent spirometry models uses the same architecture.

Parameter	***n conv***	***depth***	***filt size***	***pool size***	***n fc***
Value	4	64	7x7	2x2	3

This architecture is significantly heavier than the Confidence model as it has greater *depth*, *n conv*, *n fc* and a smaller *pool size*. In total it has 4.4 million parameters with 3.3 million trainable neurons, occupying 45 megabytes when fully trained. The four spirometry models are trained several times to ensure training converges to similar weights each time. Convergence tends to occur around 100 iterations, which takes about an 1.5 hours.

CurveNet Model

CurveNet uses the same inputs, X as ScalarNet with trainable spectrogram scaling and frequency cutoffs. The architecture parameters are also very similar, the only difference

being fewer *n fc* layers as the output doesn't need to be widdled down as much since it is a sequence. The architecture parameters are shown in Table 12.3.

Tab. 12.3: Parameters used for the CurveNet CNN model.

Parameter	***n conv***	***depth***	***filt size***	***pool size***	***n fc***
Value	4	64	7x7	2x2	0

This architecture is actually lighter weight than the Scalar model as it does not use FC layers, although the multiple outputs may slow it down. In total it has 2.7 million parameters with 1.6 million trainable neurons, occupying 30 megabytes when fully trained. Training, however, is much slower as the model must learn how to balance all of the outputs and how to share parameters useful for predicting them. Furthermore, since the output is a time-varying sequence, the CurveNet must learn how to keep track of how the sound varies with time. Typical training can last anywhere between 3 and 8 hours and usually requires 300 to 1000 iterations. It is unclear what the optimal training time is as the overall loss seems to continue to drop, well into 1000 iterations. That being said a model trained with only 200 iterations is generally only marginally worse (about 0.02 liters more in average error) on FEV1 compared to 1000 iterations. This suggests the model may begin to overfit to the data and does not learn anything new after around 200 iterations, or 3 hours. More experimentation is required to fully understand this.

CurveNet Cost Function

What differentiates CurveNet from ScalarNet is the way the output and cost function are handled. The last trainable FC layer, outputs a volume of size (500, 1, 1), corresponding to a 10 second FT curve. The FT curve is used as the main output as it represents a sequence physically similar to the input audio. One issue is the input audio is only 5 seconds long, which means the model must extrapolate to 10 seconds of flow. In most cases, the flow is nearly 0 within 5 seconds, so this is not too much of an issue. It does, however limit the predictive power for the FET metric. Since there is no ground truth for FT, nor a desire for its use by clinicians, the VT curve is derived from the FT using integration, or a scaled cumulative sum. With the FT and VT curve, all the spirometry metrics can be derived. PEF and FVC are the max value of the FT and VT curve respectively, and FEV1 is the value of the VT curve at 1 second or the 50th sample. FET is the time at which the FT decays to 0, or the time when FVC is reached. the FV curve can be assembled after the other curves are predicted as it is simply a combination of the FT and VT plots.

While these parameters are easy to derive with conventional mathematics, it is much more difficult to derive them using differentiable operations supported by Tensorflow. Nonetheless, a custom output block was eventually implemented to derive all of the curves and parameters. These derivations do not require any trainable parameters, and thus do not add much complexity to the model. As a result, the custom output block takes in a volume of *(500, 1,*

1) and outputs 6 volumes, namely: FT curve and VT curve with volume *(500, 1, 1*), as well as scalars, FEV1, FVC, PEF, and FET with volume *(1, 1, 1)*.

The custom cost function must compute a cost, or loss, as a function of all of these outputs. For each output, mean squared error is used, but the outputs are weighted differently. The weights, which sum to 1 are as followed: 0.2 for FT and VT, 0.25 for FVC, 0.3 for FEV1, 0.04 for FET and 0.01 for PEF. These weights are set somewhat subjectively as FEV1 is by far the most useful metric, followed by FVC. Experimental results show if PEF has too high of a weight, the curves begin to look unnatural as they have random impulse to try to hit the PEF at some point in the curve. Also, PEF and FET have very little clinical relevance so are treated as so. One way to think about the cost function is as a measure of similarity between the ground truth curves and the predicted curves, with a heavy penalty on the points where the spirometry metrics occur. This way, the shape of the curve is most accurate near the points that matter.

12.9 Conclusion

In summary, the generic pipelines have been implemented to support the training and evaluation of several types of models that solve a few specific purposes required for the *Spiro AI* system. These purposes include trimming effort to isolated the exhale, affirming the exhale is acceptable with a sufficient degree of confidence, and predicting the spirometry outputs to generate a report. The next chapter, Results, will cover the results of evaluating the models proposed in this chapter.

13 Results

This chapter summarizes the results of the methods proposed in Chapter 12, as well as the main insights. The Trimming models are evaluated subjectively as there is no established ground truth for the trim regions yet. The Confidence and Prediction models are evaluated using the evaluation pipeline which provides a thorough evaluation of each proposed model, including error metrics and feature importance insights. More analysis is conducted on the CurveNet results, including examples of the outputted curves.

13.1 Trimming Model

A formal evaluation of the trimming model has not been completed, however, a manual inspection of 2000 results has been performed as a sanity check.

13.1.1 Rule-based Trimming Model

The Rule-based method successfully trimmed 9/10 recordings. Figure A of 13.1 shows a correct trimming where the start is just before the impulse and the end is right before another superfluous impulse. In Figure B, the model correctly segments an exhale, but it is unclear whether it is the ideal one as there appear to be two separate valid ones. Figure C is an example of a failed trimming as the impulse at the start is a false positive and the true start happens a few seconds later.

13.1.2 Cross-Correlation Trimming Model

The Cross-Correlation trimming Model, had comparable results although appeared to be more precise at finding the start and less prone to false positive such as in Figure C of 13.1. The downside to this method is it only finds the start. A separate rule-based strategy must be implemented to find the end. For this reason, the original Rule-based method is still preferred. The reference signal, trained using unsupervised expectation maximization algorithm, is shown in Figure D of 13.1. The optimized reference signal is simply a sharp impulse with a rapid decay to 0.

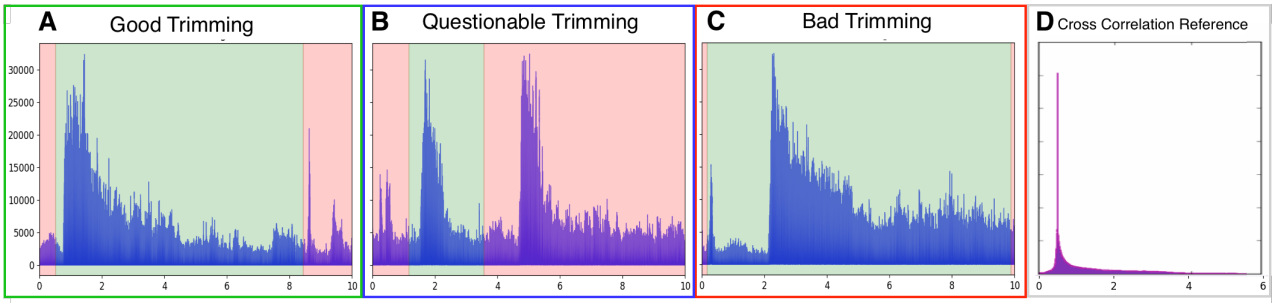


Fig. 13.1: A-C show results of the Rule-based trimming model where (A) Shows a good trim result, (B) is an example where it looks like two exhales occur and the first is successfully segmented. (C) is a failure due to the false positive impulse before the exhale. (D) Shows the cross-correlation reference signal. For all plots, the x axis is the time in seconds and y axis amplitude and the absolute value of the waveform is plotted.

13.2 Confidence Model

The evaluation results for ConfidenceNet, GBM, and Log L2 on 200 unseen efforts are organized in Table 13.1. In almost all metrics, the GBM trained on all features performs significantly better, achieving about 5% more accuracy on the evaluation set. Given only the Mel-specs as input features, ConfidenceNet slightly outperforms the competition. In the case of Log L2, it seems the Mel based features do not help much since the *no-mels* and *all* variants achieved similar performance. This suggests the Mel-spectrograms do not expose what makes up an acceptable effort in a linear way, although such information can be gathered with non-linear observations as shown in ConfidenceNet and GBM.

Tab. 13.1: The results from evaluating various models on the confidence evaluation set with 200 difficult entries.

Model	Features	Accuracy	Fscore	Precision	Recall
ConfidenceNet	only mels	0.852	0.850	0.814	0.888
GBM	only mels	0.843	0.837	0.813	0.863
Log L2	only mels	0.821	0.821	0.775	0.873
Log L2	no mels	0.845	0.847	0.789	0.914
GBM	no mels	0.876	0.874	0.837	0.914
GBM	all	**0.895**	**0.891**	**0.870**	**0.914**
Log L2	all	0.850	0.845	0.819	0.873

13.2.1 Manual Feature Importance

Analyzing the feature importance for each feature set highlights in most cases the features quantifying the explosive start of an exhale are the most useful predictors of an acceptable effort.

No-mel Feature Set

This is the simplest feature set as it contains basic waveform loudness properties as well as the polynomial coefficients and downsampled amplitude and frequency envelopes for a total of 60 features. In the case of Log L2, the features with the largest weight are the loudness of the first 200ms, followed by other features describing the beginning of the sound. The least important feature is the room loudness, which makes sense as it doesn't describe the exhale effort at all. In GBM, the list is similar, features describing the explosive part of the effort tended to have higher important, although GMB favored the low pass filtered feature variants. The least important features are the ones describing the tail end of the sound.

Only-mel Feature Set

This set consists of 80 features describing a downsampled Mel-scaled spectrogram. It is evaluated in order to compare to the NN model. Oddly enough the Log L2 and GBM are opposite in terms of the feature importance. Log L2 favors the Mel-spec regions that describe the dead space, that is, the times and frequencies where the obvious exhale qualities do not exist (end of the recording or high pitch frequencies). Conversely, GBM favors the Mel-spec regions where the exhale lives, the beginning and lower frequencies.

All Feature Set

This set combines all features for a total of 140. The order of importance is essentially the superposition of the two prior sets. In Log L2 the loudness of the first 200ms is still the largest weight, but the following features are all Mel-spec based. The GBM as in other feature sets, most utilized any features describing the explosive start.

13.2.2 Receiver Operating Characteristic Curves

The Receiver Operating Characteristic curve (ROC), shown in Figure 13.2 plots the true positive rate against the false positive rate at different decision thresholds. The plots show the trade-off between sensitivity and specificity, an increase in sensitivity will be accompanied by a decrease in specificity. The further the curve is from the diagonal line, or closer it is to the left, top corner, the more accurate the test. Given a binary classification model such as the Confidence model, one can tune the behavior to either have high detection rate, at the cost of more false positives, or sacrifice detection accuracy in order to minimize false positives. For this work, it makes sense to choose a decision threshold that maps to the elbow (top left corner) of the ROC curves since for the most accurate models, this corresponds to a $< 10\%$ false positive rate and a true positive rate $> 85\%$.

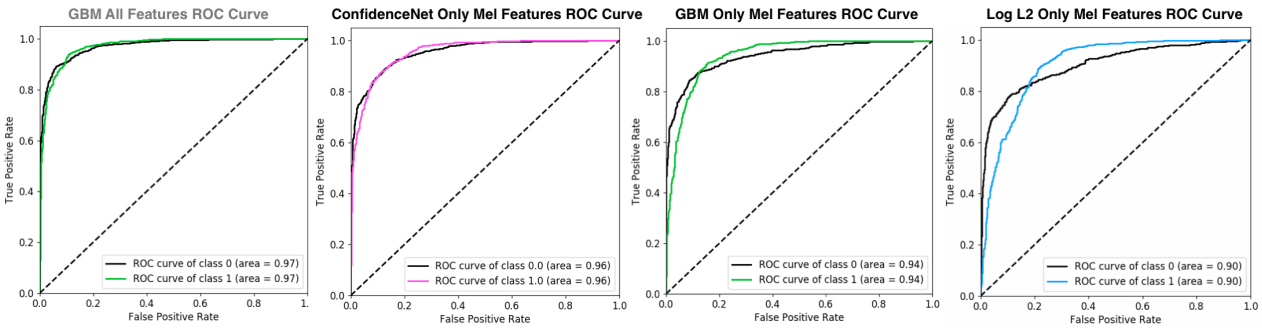


Fig. 13.2: ROC plots for (A) GBM with *all* features show a balance between the true and false positive rate in both classes, and the *Only-mel* features for (B) ConfidenceNet, (C) GBM, and (D) Log L2. From left to right the models become more vulnerable to false positives

13.3 Prediction Model

The results of evaluating ScalarNet, CurveNet GBM, and Log L2 on 772 unseen, somewhat uniformly distributed patients are organized in Table 13.2. Unlike the with Confidence models, the neural network approaches perform best. This is likely because the problem is much more complex, beyond the limits of the manually extracted features.

Tab. 13.2: The evaluation results for the spirometry Prediction models, evaluated on 772 uniformly distributed entries. The error metric used is absolute error.

Model	Features	FEV1	FVC	PEF	FET
Mean Tracking	NA	0.78	0.68	1.90	1.73
CurveNet	only mels	**0.48**	**0.50**	1.39	1.72
ScalarNet	only mels	0.50	0.52	**1.33**	1.79
Lin L2	only mels	0.63	0.62	1.69	1.70
GBM	only mels	0.56	0.54	1.51	1.69
Lin L2	no mels	0.63	0.60	1.69	1.66
GBM	no mels	0.54	0.56	1.51	**1.59**
Lin L2	all	0.60	0.57	1.69	1.65
GBM	all	0.52	0.54	1.53	1.61

Recall, FEV1 is typically between 0 and 5 L, FVC between 1 and 8 L. PEF (L/s) and FET (s) vary much more up to around 12 these insights are made apparent in Figure 11.7. The relative difficulty of predicting spirometry outputs is proportional to the variance of the distribution, as captured by the Mean Tracking model. The FET metric has proven difficult to predict as in most models the absolute error is comparable to just guessing the mean. This is likely due to two reasons: First, the audio is often trimmed to around 5 seconds, and the FET which captures the elapsed time of the exhale, is typically > 10 seconds, so the models must extrapolate in time in order to predict FET. Second, the FET is highly variable and is not well correlated with other spirometry metrics. Since the FET is mostly used clinically to screen for errors such as early stop, it is not very important as a diagnostic output and therefore can

be omitted without much of a sacrifice to the end use case. In CurveNet, the FET output is assigned a very low penalty in the cost function for these reasons. The other metrics perform far better than guessing the mean and have much more clinical relevance as well.

13.3.1 Manual Feature Importance

In the manual predictive models, four spirometry metrics are evaluated with three different feature sets for both GBM and Log L2. This results in 24 independent models. Rather than exhaustively covering the feature importance for each model, high-level observations will be offered the three features sets. In general, ML models using the *only-mel* features performed worse than the *all* or *no-mel* features, which indicates the manually defined features sets provide more predictive power than the downsampled Mel-spectrogram.

No-mel Feature Set

In the *no-mel* features, the FET and FVC metrics highly rely on the overall area of the sound. PEF and FEV1 unsurprisingly rely on the features that describe the explosive region of the sound, such as the loudness, polynomial coefficients, and the amplitude/frequency envelope. In general, FEV1 and FVC seem to use more of the features and thus assign a higher importance to most of them, while FET and PEF mainly rely on a select few. This suggests the selected features are not adequate for FET and PEF predictions.

Only-mel Feature Set

The time regions of the Mel-spectrograms found most useful for each metric is highly correlated to the time region on the flow versus time curve where each metric can be derived, which suggests the Mel-spectrogram is a valid proxy for a flow versus time curve. Therefore, the PEF and FEV1 metrics most utilized the first second of the spectrogram. FET typically utilized the start and end regions, and FVC seems to be based on the area of dead space frequency bands where the exhale does not occur, similar to the Confidence model. Unlike the *no-mel* case, all spirometry metrics tend to use all the Mel features to some degree which implies the predictions are a function of the complete spectrogram.

All Feature Set

When *all* features are used, it becomes clear that the manually defined features are preferred by the models, especially in the Log L2. In general, the most important features resemble those described in the *no-mel* section, except with a few Mel based features sprinkled in. In particular, the Mel features describing the explosive region of the exhale are valued.

13.3.2 Bland-Altman Plots

To evaluate the agreement between the proposed Predictive models and the gold standard ground truth spirometry data, a Bland-Altman (BA) plot, is used (known as a Tukey Mean-Difference Plot outside of the medical field). While correlation quantifies the strength of the linear relationship between two variables, the limits of agreement highlight the differences between to methods. A BA plot is a graphical method to plot the difference scores of two measurements against the mean for each subject [9]. Predictions for each patient appear as dots that align with the ground truth value via the x axis. The y axis highlights the error or difference between the ground truth and the prediction. The three horizontal lines represent mean of difference, called bias (the middle line, ideally at y=0) and the other two lines are limits of agreement set to plus or minus 1.96 the standard deviation.

The plot is solely meant to define the intervals of agreements and does not say whether those limits are acceptable or not. Acceptability limits must be defined separately, usually in a problem specific manner. While these plots alone do not necessarily indicate if a proposed method is passing or not, they do provide a great deal of information on how it performs on a population. Ideally, the measurement error of the proposed method is low and unbiased, this is characterized by a BA plot as randomly scattered points around the bias line that are independent of the x axis. A poor model that simply predicts the mean would appear as a linear trend increasing as the x axis increases since the error would be most negative at low x values and most positive at high x values. A BA plot also shows how uniformly distributed the sample population is based on the spread in the x direction. This allows outlier analysis to be performed as the error for outliers is encoded in the plot.

Ideally, a BA plot has all of the points within the limits of agreement and randomly distributed in the y direction around the bias line, no matter what x value. Furthermore, the span of the points along the x axis should adequately represent the population for the metric being evaluated and not be overly clustered in a narrow range. Figure 13.3 shows the Bland-Altman plots for the *only-mel* predictive models where Figure A is the FEV1 metric and B, FVC.

Clearly, there is an undesired error distribution represented in all plots as there is a weak linear trend showing overestimation for low values and underestimation for high values, especially in the Log L2 models. These plots point out models such as Log L2 that learn to predict the close to the mean. Furthermore, the plots show most of the data is concentrated in the lower region of the x axis, which re-enforces the data imbalance issues covered in Section 11.5. The evaluation set is crafted to be more evenly distributed than randomly sampling, however, there are not enough healthy patients to evenly represent the higher x axis values. All things considered, the green CurveNet BA plots demonstrate tighter

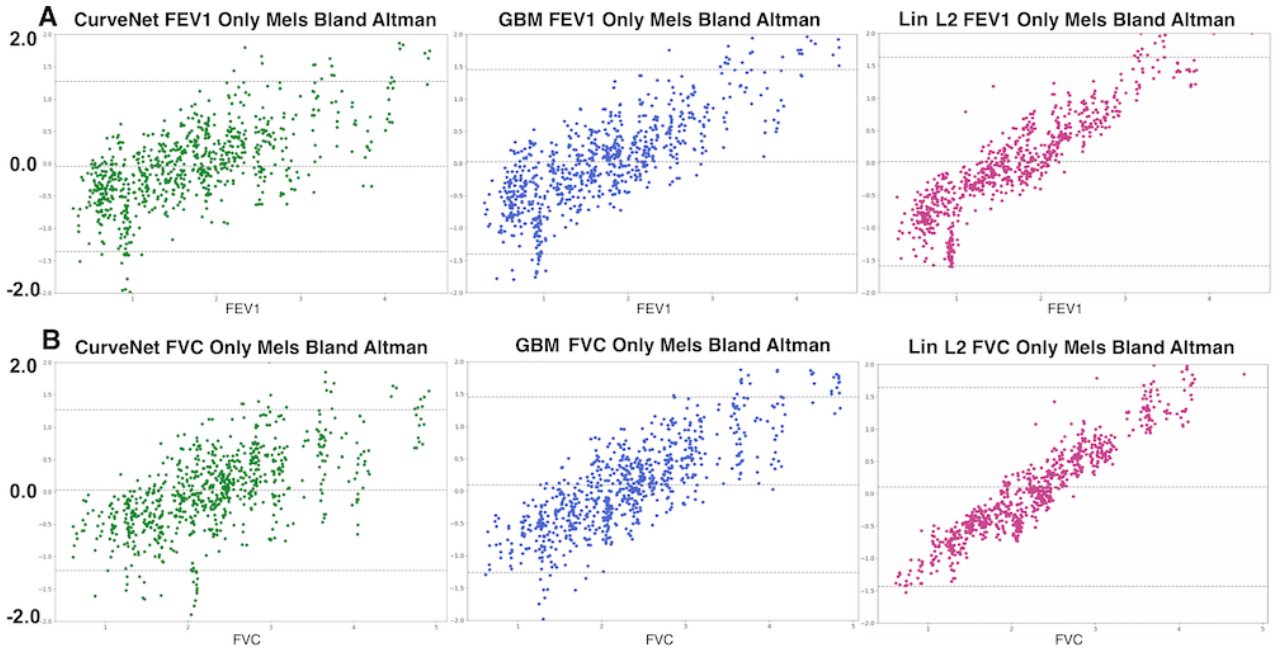


Fig. 13.3: Bland-Altman plots for (A) FEV1 and (B) FVC metric evaluating the CurveNet model (green), GBM model (blue) and Log L2 (purple) on a sample size of 772. Some of the Log L2 points are outside the scale shown.

agreement with the ground truth and a more unbiased error. This, coupled with its low absolute error as shown in Table 13.2 makes it the strongest model of the Predictive models proposed.

13.3.3 CurveNet Model

Aside from being the top performing Predictive model, CurveNet also offers much more diagnostic power is it outputs the spirometry curves in addition to the metrics. This section evaluates the error and correlation between the predicted and ground truth curves. Table 13.3 shows the curve evaluation results for the flow (FT) and volume (VT) versus time curves, where flow is measured in L/s and volume, L. The flow volume curve (FV) can be thought of as a compounded error of the FT and VT as it combines the two plots.

Tab. 13.3: The spirometry curve evaluation results specific to CurveNet for flow versus time (FT) and volume versus time (VT).

Curve	Abs Error	Corr Coeff
FT	0.14	0.87
VT	0.48	0.75

The FT has a deceivingly low absolute error because the curve is ten seconds long but mostly made up of zeros for the last 5 seconds. VT has a higher error because the final value is the

FVC rather than 0 so it is harder to fit to as FVC varies significantly per person. The full page Figure 13.6 at the end of the chapter demonstrates outputted curves (blue) overlaid with the ground truth curve (green) to visually show the correlation between randomly selected good and bad fits. As shown, it is possible to have a very tight fit and the overall error is based on how well correlated the FT curve is in the first few seconds since the other curves are derived from it. Further research will need to explore clever methods for verifying the scaling of the curves. Often time the largest error comes from scaling the PEF incorrectly as the general curve shape is often accurate.

With CurveNet, it is possible to analyze the structure of the neural network in order to expose the regions of the input that are most useful in generating the output by looking at the feature maps as the input is cascaded through the layers. This concept is similar to manual feature importance but results in a much higher resolution understanding of what the model uses in a spatial sense. Figure 13.4 summarizes this concept and represents a heat map showing the most important regions in the input Mel-spectrogram. The result reinforces some of the discussion in the Manual Feature Importance section as it is clear the explosive region is the strongest predictor of the output. Other interesting observations indicate certain frequency regions which tend to have more useful information or exist for longer durations of time. It appears the neural network tracks the decay of the sounds existing around 1.5kHz, and less so for sounds in the proximity of 3.5kHz. Furthermore, the low frequencies (between 100Hz and 300Hz) are shown to have a significant level of focus across the entire duration. It is also clear that frequencies above 4kHz, especially after the first second, are not very useful.

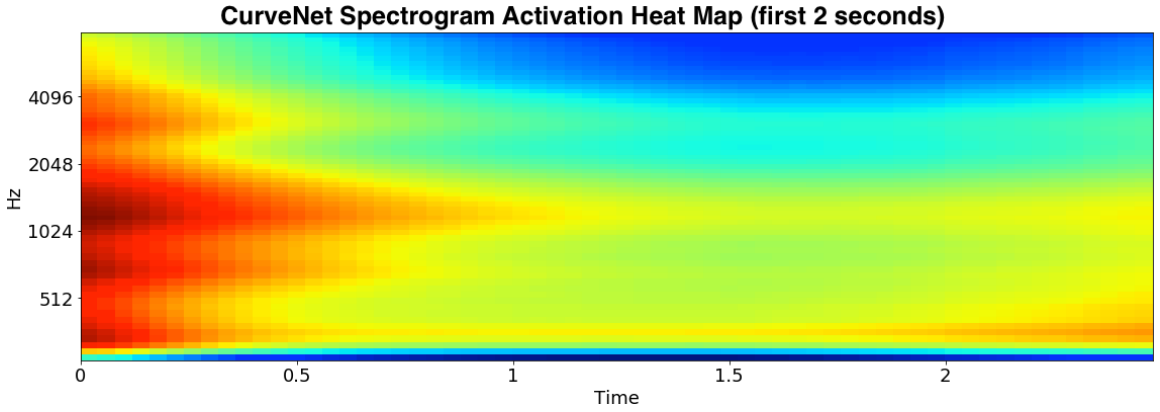


Fig. 13.4: A heat map highlighting the region of interest in the first two seconds of an input Mel-spectrogram. Generated by compounding the feature maps of various acceptable input exhale efforts.

This analysis is very useful in understanding the physical properties of a human-powered exhale, and it informs the best form of scaling the frequency axis for a spectrogram. Clearly, the Mel-spectrogram is on the right track as it allocates progressively higher resolution to the lower frequency bands, but it is certainly possible that a better frequency scale or filterbank

exists for the purposes of spirometry. For this reason, the neural network architecture has been revised to allow training a custom filterbank. While results have not yet been obtained, the hypothesis is that an ideal scaling will offer a higher resolution in the bands that are more attended to in the neural network, such as the 100Hz to 300Hz range as well as the 1.5kHz and 3.5kHz regions.

13.4 Conclusion

The results suggest each model required in the system will benefit from a different type of model, namely, the Trimming model is best suited with a fully defined Rule-based model, the Confidence model works best with GBM using *all* manual input features and the Prediction model benefits most from the CurveNet neural network. These findings are summarized in the annotated *Spiro AI* system diagram from Chapter 12 shown in Figure 13.5.

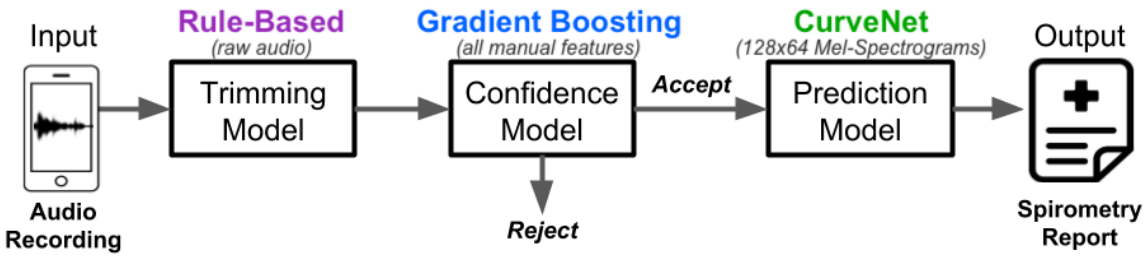


Fig. 13.5: *Spiro AI* end-to-end system with the best performing models annotated above each block.

Prior work outlined in Section 9.2 report results in percent error and evaluate on much smaller, more uniform population distributions. This work uses absolute error as a metric because percent error is misleading when sample distributions between opposing methods do not overlap. For example, an FEV1 with absolute error of 0.5 L taken from a distribution centered around an FEV1 of 4 L (as in SpiroSmart) would have a percent error of 12.5% , while the same absolute error measured from a distribution centered around an FEV1 of 2 L (as in this work) would report double the percent error (25%). This, coupled with the drastically different evaluation sizes and distributions, unfortunately, make comparing to prior work somewhat of a lost cause. Either way, the requirements set in stone by the Food and Drug Administration (FDA) require FEV1 absolute error less than 0.2 L for the device to be considered clinically adequate, so there is still a long way to go.

In summary, the results presented are far from meeting the clinical gold standard and difficult to compare to prior work, but they provide a strong baseline from which improvements can surely stem from. Given the non-ideal state of the dataset and the difficulty of the problem, these results serve as an exploration of different input features and types of models rather than an evaluation of a clinical grade technique.

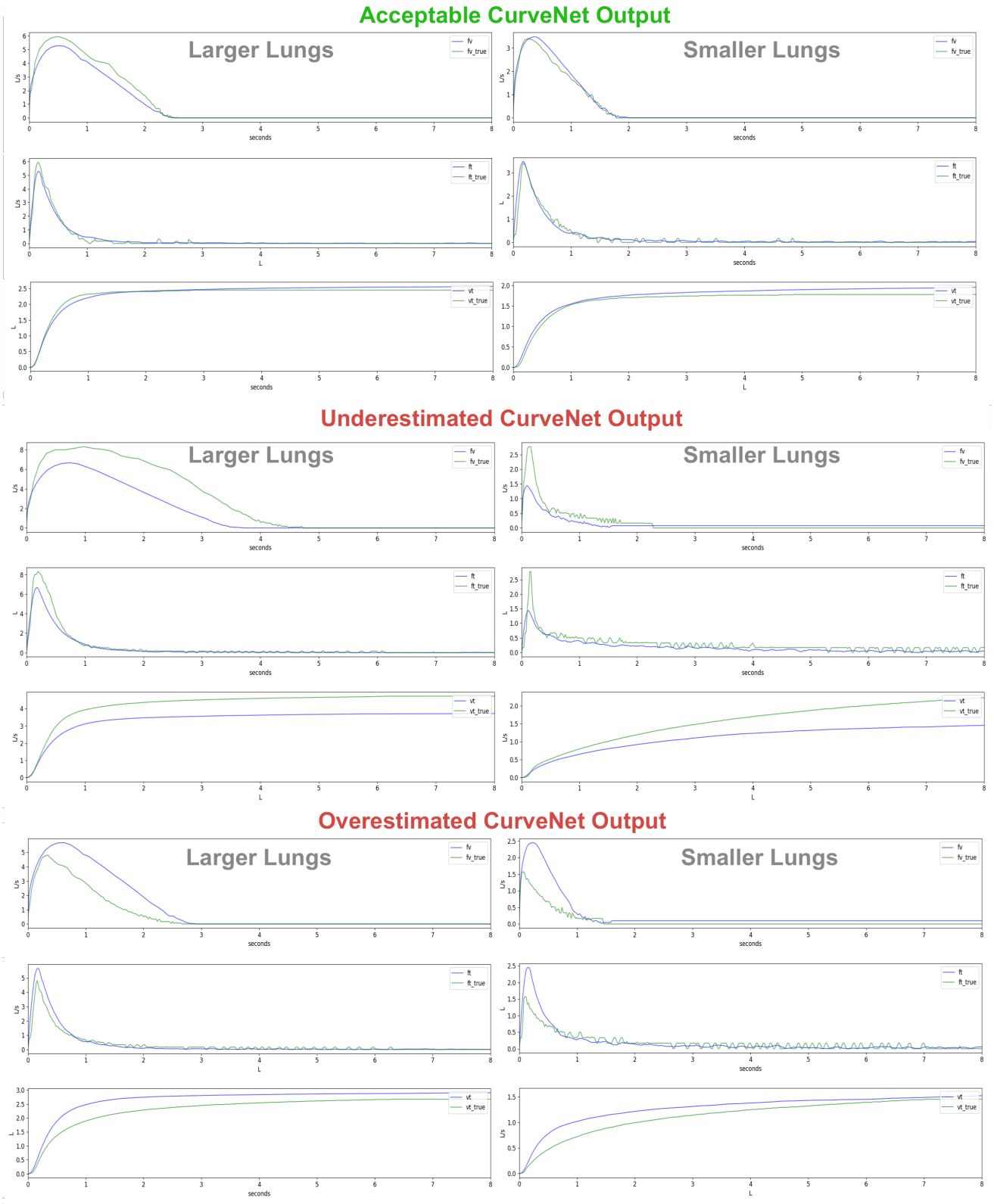


Fig. 13.6: Randomly selected examples of CurveNet curve outputs which show acceptable output as well as erroneous outputs for large and small lung size

14 Deployment

In order to support future research, data collection, user experience studies and live demonstrations, a backend server, and iOS app have been developed and are outlined in the following sections. This work is a collaborative effort involving many more individuals than the author of this work.

14.1 Spiro AI Backend Server

The original SpiroSmart backend relied on Matlab and several outdated libraries which over time, proved to be unreliable and non-ideal for a production quality experience. Recently the server has been rebuilt with modern tools such as Docker as well as the models and preprocessing pipeline outlined in Chapter 12, which are optimized for fast, parallel computation, compared to Matlab. The new backend is complete with tests and reproducibility metrics to encourage other developers are equipped to collaborate and contribute to the codebase. All of the preprocessing and model execution can either be run on a CPU or GPU and are fast enough to the point where the audio file upload time is the biggest bottleneck. Furthermore, there are debug options to permit viewing the source audio files and output spirometry report in test scenarios. Finally, it is fairly straightforward to deploy new models, adjust the logic, or migrate to another server thanks to Docker and Git integration.

14.2 FreshAir iOS

To serve as a frontend to the server backend, an iOS app named FreshAir has been developed and deployed to a number of global clinics. Aside from facilitating easy testing and experimentation, the FreshAir app also allow usability studies to be conducted, as well as live demonstrations. In fact, the original purpose of it was to explore different user experience strategies to help educate and instruct users on performing an acceptable spirometry effort. Through tight collaboration with the Spirometry 360 organization and the Seattle Children's Hospital, the app was deployed to several clinics in Europe and Asia and has been used for other research and data collection for nearly a year.

There have also been a number of high profile live demonstrations for various tech CEOs, researchers and according to our Russian collaborator, even the Russian Prime Minister, Dmitry Medvedev, has seen a demonstration. The FreshAir app has been a great vehicle for visibility and also provides an avenue for people unfamiliar with health or machine learning research to get excited about the mobile health revolution. Currently, the app supports all of the models and outputs described in this work and is used as follows:

1. The user or clinician to enters patient information, or select an already existing one.
2. At this point, patient trends or historical data can be viewed if permission has been granted.
3. The user is walked through how to perform the maneuver with several languages and even a video.
4. Once the user is ready, they can begin the first trial by tapping the *OK, Let's take the test!* button.
5. Now, the phone streams the front-facing camera to act as a mirror so the user can verify the distance and alignment matches the instructions, as well as visualize their effort as they perform it.
6. When the user taps *Start*, a countdown begins and the user must inhale, then exhale when the countdown terminates.
7. The app records the audio for 8 seconds, then automatically uploads it to the backend server. The system proposed in 12.1 preprocesses the audio, checks the quality of the exhale, then if accepted computes the spirometry report.
8. The result is sent back to the phone and the uploaded audio is saved for future algorithm development (per IRB).
9. If the effort is rejected or the internet connection is not working, the user is prompted with a descriptive error.
10. Otherwise, the Results screen shows the latest trial spirometry report with complete with curves and percent predicted metrics.
11. To complete the session, the user must submit more trials until the reproducibility criteria is met (see Section 5.4).
12. At any point, the user can inspect the volume versus time (VT) or flow versus volume curve (FV) for the current and all previous trials in the session.
13. Once reproducibility is achieved, or the user prematurely taps *Finish Session* the Final Results screen displays the level of reproducibility as well as the best effort, which is stored as the result for that session. If the result is not reproducible, the user can go back and continue doing more trials.

All of the IDs (see Chapter 11), including patient, session, clinic, and phone ID are handled automatically by the backend and frontend in an organized manner so the clinician or user

does not need to keep track of anything. The app screens, numbered based on the above step they refer to, are shown in Figure 14.1.

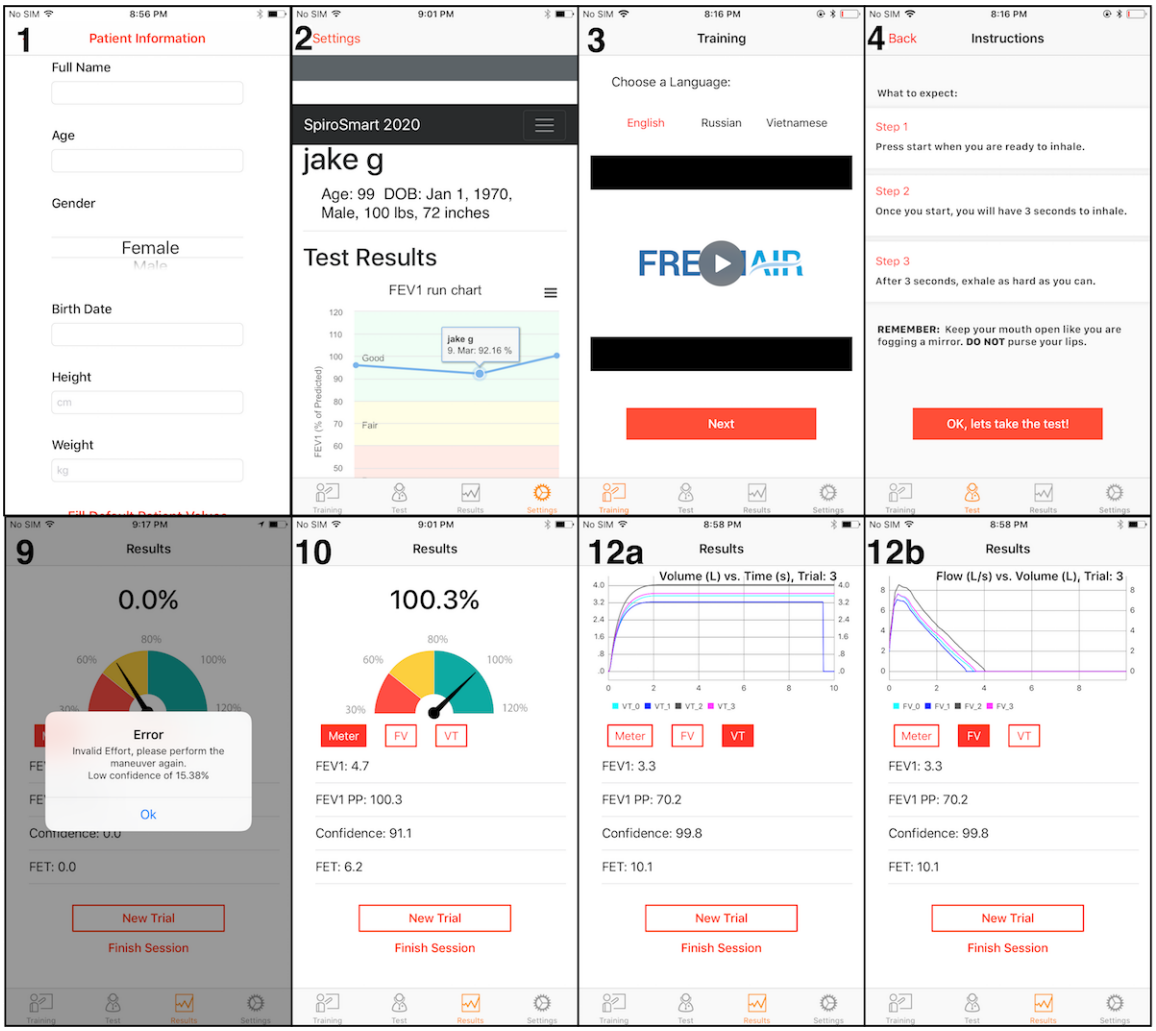


Fig. 14.1: Screenshots taken from the FreshAir iOS app with the step number from the list above annotated.

Future Updates

FreshAir is still in its infancy and is well poised for many updates. For example, a much stronger quality control (QC) framework is being developed. Rather than simply accepting and rejecting audio efforts, the future Confidence model will include capabilities for classifying the spirometry error and offering feedback. The QC framework will also utilize the front facing camera to get a sense for the phone to mouth distance and the diameter of the user's mouth during the effort. This will help educate the users on the correct maneuver and perhaps allow the audio to be normalized based on the phone to mouth distance. Metrics such as FET and errors such as early stop, which is difficult to obtain via audio, can be measured using a vision based model as well.

Another planned update is to move the Confidence and Prediction models to the device. This will greatly speed up the turnaround side, and allow users to feel more comfortable as they can opt out of sending audio to the server without losing the main functionality of the app. This would be much more feasible if the Trimming and Confidence models are condensed into a single neural network architecture such that both the Confidence and Prediction models require only Mel-spectrograms as input features. There are several successful examples demonstrating neural networks running very well on an iOS device, and now Apple has launched Core-ML which makes the process even easier.

Conclusion

All in all, the app and accompanying backend allow for rapid end to end development, live demonstrations, and user studies. Furthermore, the app gives the research great visibility which inspires others the invest in future mobile health endeavors. Eventually, this app may be accessible to those who actually need it for health reasons.

15 Conclusion

The main objective of this work has been to investigate the effectiveness of various machine learning strategies in predicting spirometry metrics from the sound of a forced expiratory effort. The investigation branched into three separate paths, each with the goal of estimating spirometry metrics through a sound based model. First, physics and fluid dynamics based models were derived with the hopes of explicitly modeling airflow. This method proved to be limited as there are several unknowns and uncontrollable variables at play. It was then postulated that the unknowns could be effectively understood through data-driven modeling techniques. The second approach comprised of computing manual features from the audio and using classical machine learning to tune a mapping from the extracted features to the desired outcome. This approach naively assumes the manual extracted features are adequate for the problem at hand. Up to this point, the approaches were based on prior research in mobile spirometry and airflow modeling. The third approach explored cutting-edge deep learning techniques to devise a neural network which automatically extracts features from the exhale sound that are ideal for modeling the intended output.

The latter two machine learning approaches proved to be useful in two of the subproblems defined in the end to end Spiro AI system, namely the Prediction model and Confidence model. The Prediction model, being the most complex subsystem, benefited most from the deep learning architecture, specifically the CurveNet model which was devised specifically for outputting a flow versus time curve from which all relevant spirometry metrics can be derived. In contrast, the Confidence model, which was designed to reject incorrect efforts based sound, performed best using the expert defined manual features. Finally, the Trimming model which simply extracts the region of sound where the exhale occurs performed best using an explicitly defined Rule-based method, although this subproblem was not thoroughly evaluated. Anticlimactically, the deep learning approaches, while a million times more complex were certainly not a million times more effective. This suggests there are inherent limitations set in place by the dataset, problem complexity and signal to noise ratio.

A significant contribution outside of the model evaluations and the end to end system lies in the cleansing and organization of the massive global dataset. Through this process, the collection and organization procedures were found to be less than ideal. As a result, the quality of the data was sacrificed along with over half of the 40k entries. Following the organization, the statistics of the set were analyzed and a large bias towards unhealthy

abnormal lung function was revealed. The dataset, despite its flaws and limitations, was still incredibly valuable as it allowed various machine learning techniques to be evaluated and compared. While the resulting models are not ready for the real world until a more thoughtful and balanced data collection effort is completed, the main conclusions and proposed system will no doubt have a lasting effect on future work.

Aside from the models that serve as the main components in the complete Spiro AI system, generic machine learning pipelines and experimental procedures for tasks such as ultrasonic sensing were documented and open sourced for use in other problem areas. Furthermore, a backend server and iOS app were developed and deployed for use in future data collection efforts and demonstrations.

In summary, this work proposed a deployable end to end solution which is trained and evaluated on the largest known spirometry dataset and is the first to effectively use deep learning to model the relationship between sound and human-powered airflow. While the results suggest the problem is far from being solved with respect to what regulatory agencies such as the FDA define as acceptable, spirometry is now closer to a smartphone based solution than ever before. The next section will outline the work necessary for advancing this research closer to a solution that has the potential to end up in over 2 billion people's pockets.

15.1 Future Work

Much of the future work has been alluded to throughout the chapters and can be categorized into future studies and data collection efforts as well as future technical developments.

Data Collection

- A highly regulated and organized global data collection effort focused on collecting smartphone-based sound recordings and spirometry ground truth from a balanced distribution with respect to health, age, ethnicity, and gender. This can be used to improve and validate the Sprio AI models on a more realistic distribution.
- A well defined longitudinal study on a handful of individuals with existing lung conditions such as COPD or asthma to evaluate the effectiveness of trend reporting and treatment monitoring.
- A qualitative longitudinal study on volunteer smokers that may be trying to quit and are willing to try smartphone based spirometry as a motivational feedback system for assessing lung function.
- Evaluation using the FDA appointed ATS waveform generator which is an airflow generator used to validate clinical spirometers.

Technical Advancements

- A more complete confidence model with built-in trimming capabilities and automatic spirometry error classification based on the sound of the effort.
- Deeper exploration into neural network architectures, particularly CRNNs.
- Further exploration into more advanced airflow physics models using software simulations and complex fluid models.
- Implementation of the Spiro AI system on a smartphone rather than in the cloud in order to preserve user's privacy and enable usage in regions lacking high-speed internet.

The work completed so far, as well as the proposed future work is clearly multidisciplinary and will require collaboration between physicians, physicists, engineers, regulatory committees, computer scientists, and volunteer patients to name a few. It is the collaborative quality that makes this type of work inspiring and impactful for many, yet treacherous and daunting for others.

Bibliography

[1] Ahmad Abushakra and Miad Faezipour. “Lung capacity estimation through acoustic signal of breath”. In: *Bioinformatics & Bioengineering (BIBE), 2012 IEEE 12th International Conference on*. IEEE. 2012, pp. 386–391 (cit. on p. 112).

[2] Lara J Akinbami, Cathy M Bailey, Carol A Johnson, et al. “Trends in asthma prevalence, health care use, and mortality in the United States, 2001-2010”. In: (2012) (cit. on p. 32).

[3] *Apple Products Are Driving Market Growth for MEMS Microphones IHS Says* (cit. on p. 61).

[4] Philippe Arlotto, Michel Grimaldi, Roomila Naeck, and Jean-Marc Ginoux. “An ultrasonic contactless sensor for breathing monitoring”. In: *Sensors* 14.8 (2014), pp. 15371–15386 (cit. on pp. 78, 113).

[5] *Asthma*. Nov. 2013 (cit. on p. 26).

[6] Bethany B. Moore, William E Lawson, Tim D Oury, et al. “Animal models of fibrotic lung disease”. In: *American journal of respiratory cell and molecular biology* 49.2 (2013), pp. 167–179 (cit. on p. 23).

[7] Timothy J Barreiro, Irene Perillo, et al. “An approach to interpreting spirometry”. In: *American family physician* 69.5 (2004), pp. 1107–1116 (cit. on pp. 49, 54).

[8] Henry E Bass, Richard Raspet, and John O Messer. “Experimental determination of wind speed and direction using a three microphone array”. In: *The Journal of the Acoustical Society of America* 97.1 (1995), pp. 695–696 (cit. on p. 113).

[9] J Martin Bland and DouglasG Altman. “Statistical methods for assessing agreement between two methods of clinical measurement”. In: *The lancet* 327.8476 (1986), pp. 307–310 (cit. on p. 160).

[10] Brigitte M Borg, Moegamat Faizel Hartley, Mo T Fisher, and Bruce R Thompson. “Spirometry training does not guarantee valid results”. In: *Respiratory care* 55.6 (2010), pp. 689–694 (cit. on p. 42).

[11] Stuart Bradley, Juha Backman, Sabine von Hunerbein, and Tao Wu. “The mechanisms creating wind noise in microphones”. In: *Audio Engineering Society Convention 114*. Audio Engineering Society. 2003 (cit. on pp. 76, 113).

[12] Alwin FJ Brouwer, Ruurd Jan Roorda, and Paul LP Brand. “Home spirometry and asthma severity in children”. In: *European Respiratory Journal* 28.6 (2006), pp. 1131–1137 (cit. on pp. 38, 44).

[13] David C Catling, Christopher R Glein, Kevin J Zahnle, and Christopher P McKay. “Why O2 Is Required by Complex Life on Habitable Planets and the Concept of Planetary" Oxygenation Time"”. In: *Astrobiology* 5.3 (2005), pp. 415–438 (cit. on p. 7).

[14] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, et al. “Learning phrase representations using RNN encoder-decoder for statistical machine translation”. In: *arXiv preprint arXiv:1406.1078* (2014) (cit. on p. 108).

[15] Jinwook Choi, Sooyoung Yoo, Heekyong Park, and Jonghoon Chun. “MobileMed: a PDA-based mobile clinical information system”. In: *IEEE Transactions on Information Technology in Biomedicine* 10.3 (2006), pp. 627–635 (cit. on p. 111).

[16] Benoit Cushman-Roisin. “What is Environmental Fluid Mechanics?” In: *Environmental Fluid Mechanics* 1.1 (2001), pp. 1–2 (cit. on p. 80).

[17] Lilian De Greef, Mayank Goel, Min Joon Seo, et al. “Bilicam: using mobile phones to monitor newborn jaundice”. In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. 2014, pp. 331–342 (cit. on p. 111).

[18] Laura Dwyer-Lindgren, Amelia Bertozzi-Villa, Rebecca W Stubbs, et al. “Inequalities in life expectancy among US counties, 1980 to 2014: temporal trends and key drivers”. In: *JAMA internal medicine* 177.7 (2017), pp. 1003–1011 (cit. on p. 31).

[19] Laura Dwyer-Lindgren, Amelia Bertozzi-Villa, Rebecca W Stubbs, et al. “Trends and patterns of differences in chronic respiratory disease mortality among US counties, 1980-2014”. In: *Jama* 318.12 (2017), pp. 1136–1149 (cit. on p. 31).

[20] E Falaschetti, J Laiho, P Primatesta, and S Purdon. “Prediction equations for normal and low lung function from the Health Survey for England”. In: *European Respiratory Journal* 23.3 (2004), pp. 456–463 (cit. on pp. 50, 130).

[21] Flurrymobile. *Health and Fitness App Users Are Going the Distance with Record High Engagement*. Sept. 2017 (cit. on p. 40).

[22] Stanley S Franklin, Lutgarde Thijs, Tine W Hansen, Eoin O’brien, and Jan A Staessen. “White-coat hypertension: new insights from recent studies”. In: *Hypertension* 62.6 (2013), pp. 982–987 (cit. on p. 44).

[23] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, et al. “Audio set: An ontology and human-labeled dataset for audio events”. In: *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE. 2017, pp. 776–780 (cit. on p. 114).

[24] Henry W Glindmeyer, Sharon T Anderson, John E Diem, and Hans Weill. “A comparison of the Jones and Stead-Wells spirometers”. In: *Chest* 73.5 (1978), pp. 596–602 (cit. on p. 36).

[25] *Global smartphones sales revenue 2013-2017 | Statistic* (cit. on p. 39).

[26] Oleg A Godin, Vladimir G Irisov, and Mikhail I Charnotskii. “Passive acoustic measurements of wind velocity and sound speed in air”. In: *The Journal of the Acoustical Society of America* 135.2 (2014), EL68–EL74 (cit. on p. 113).

[27] S.E. Gould. *The origin of breathing: how bacteria learnt to use oxygen*. July 2012 (cit. on p. 9).

[28] Anthony J Guarascio, Shauntá M Ray, Christopher K Finch, and Timothy H Self. “The clinical and economic burden of chronic obstructive pulmonary disease in the USA”. In: *ClinicoEconomics and outcomes research: CEOR* 5 (2013), p. 235 (cit. on p. 44).

[29] Siddharth Gupta, Peter Chang, Nonso Anyigbo, and Ashutosh Sabharwal. “MobileSpiro: Portable open-interface spirometry for Android”. In: *Proceedings of the 2nd Conference on Wireless Health*. ACM. 2011, p. 24 (cit. on pp. 39, 112).

[30] Sidhant Gupta, Daniel Morris, Shwetak Patel, and Desney Tan. “Soundwave: using the doppler effect to sense gestures”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2012, pp. 1911–1914 (cit. on p. 117).

[31] Siobhan K Halloran, Anthony S Wexler, and William D Ristenpart. “Turbulent dispersion via fan-generated flows”. In: *Physics of Fluids* 26.5 (2014), p. 055114 (cit. on p. 80).

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778 (cit. on p. 97).

[33] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780 (cit. on p. 108).

[34] Heinrich D Holland. “The oxygenation of the atmosphere and oceans”. In: *Philosophical Transactions of the Royal Society of London B: Biological Sciences* 361.1470 (2006), pp. 903–915 (cit. on p. 8).

[35] *How the Lungs Work* (cit. on pp. 12, 13, 22, 26, 27, 47).

[36] Anastasia F Hutchinson, Anil K Ghimire, Michelle A Thompson, et al. “A community-based, time-matched, case-control study of respiratory viruses and exacerbations of COPD”. In: *Respiratory medicine* 101.12 (2007), pp. 2472–2481 (cit. on p. 44).

[37] Sergey Ioffe and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *arXiv preprint arXiv:1502.03167* (2015) (cit. on p. 100).

[38] Melvin Johnson, Mike Schuster, Quoc V. Le, et al. *Google’s Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation*. Tech. rep. Google, 2016 (cit. on p. 84).

[39] WC Jones. “Condenser and carbon microphones—their construction and use”. In: *Journal of the Society of Motion Picture Engineers* 16.1 (1931), pp. 3–22 (cit. on p. 62).

[40] Romain Kessler, Elisabeth Ståhl, Claus Vogelmeier, et al. “Patient understanding, detection, and experience of COPD exacerbations: an observational, interview-based study”. In: *Chest* 130.1 (2006), pp. 133–142 (cit. on p. 44).

[41] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014) (cit. on p. 99).

[42] Ronald J Knudson, Ronald C Slatin, Michael D Lebowitz, and Benjamin Burrows. “The maximal expiratory flow-volume curve: normal standards, variability, and effects of age”. In: *American Review of Respiratory Disease* 113.5 (1976), pp. 587–600 (cit. on p. 50).

[43] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, pp. 1097–1105 (cit. on pp. 95, 104).

[44] Jiri Kroutil and Miroslav Husak. “Detection of breathing”. In: *Advanced Semiconductor Devices and Microsystems, 2008. ASDAM 2008. International Conference on*. IEEE. 2008, pp. 167–170 (cit. on p. 112).

[45] Tsung-Ting Kuo, Hyeon-Eui Kim, and Lucila Ohno-Machado. “Blockchain distributed ledger technologies for biomedical and health care applications”. In: *Journal of the American Medical Informatics Association* 24.6 (2017), pp. 1211–1220 (cit. on p. 42).

[46] Eric C Larson, Mayank Goel, Gaetano Boriello, et al. “SpiroSmart: using a microphone to measure lung function on a mobile phone”. In: *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM. 2012, pp. 280–289 (cit. on pp. 3, 78, 112).

[47] Eric C Larson, TienJui Lee, Sean Liu, Margaret Rosenfeld, and Shwetak N Patel. “Accurate and privacy preserving cough sensing using a low-cost microphone”. In: *Proceedings of the 13th international conference on Ubiquitous computing*. ACM. 2011, pp. 375–384 (cit. on p. 111).

[48] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324 (cit. on p. 104).

[49] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations”. In: *Proceedings of the 26th annual international conference on machine learning*. ACM. 2009, pp. 609–616 (cit. on p. 106).

[50] A Léger. “Strategies for remote detection of life”. In: *Planets Outside the Solar System: Theory and Observations*. Springer, 1999, pp. 397–412 (cit. on p. 8).

[51] Karel F Liem. “Form and function of lungs: the evolution of air breathing mechanisms”. In: *American Zoologist* 28.2 (1988), pp. 739–759 (cit. on pp. 10, 12, 16).

[52] Bjørn Lomborg. “Global problems, local solutions: Costs and benefits”. In: *Cambridge University Pres* 143 (2013) (cit. on p. 24).

[53] *Lung Disease Respiratory Health Center* (cit. on p. 22).

[54] Andrew Z Luo, Eric Whitmire, James W Stout, Drew Martenson, and Shwetak Patel. “Automatic characterization of user errors in spirometry”. In: *Engineering in Medicine and Biology Society (EMBC), 2017 39th Annual International Conference of the IEEE*. IEEE. 2017, pp. 4239–4242 (cit. on pp. 42, 113).

[55] Nadim Maluf. *An introduction to microelectromechanical systems engineering*. 2002 (cit. on p. 61).

[56] Alex Mariakakis, Megan A Banks, Lauren Phillipi, et al. “Biliscreen: smartphone-based scleral jaundice monitoring for liver and pancreatic disorders”. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1.2 (2017), p. 20 (cit. on p. 111).

[57] Warren S McCulloch and Walter Pitts. “A logical calculus of the ideas immanent in nervous activity”. In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133 (cit. on p. 94).

[58] John A McDonald, EJ Douze, and Eugene Herrin. “The structure of atmospheric turbulence and its application to the design of pipe arrays”. In: *Geophysical Journal International* 26.1-4 (1971), pp. 99–109 (cit. on pp. 77, 113).

[59] Umberto Melia, Felip Burgos, Montserrat Vallverdú, et al. “Algorithm for automatic forced spirometry quality assessment: technological developments”. In: *PloS one* 9.12 (2014), e116238 (cit. on p. 113).

[60] Noh Mijin, Hyeongyu Jang, Beomjin Choi, and Gantumur Khongorzul. “Attitude toward the use of electronic medical record systems: exploring moderating effects of self-image”. In: *Information Development* (2017), p. 0266666917729730 (cit. on p. 42).

[61] Martin Raymond Miller, JATS Hankinson, V Brusasco, et al. “Standardisation of spirometry”. In: *European respiratory journal* 26.2 (2005), pp. 319–338 (cit. on pp. 50, 55, 57, 58).

[62] M Minksy and S Papert. “Perceptrons: An introduction to computation geometry”. In: *MIT press* 200 (1969), pp. 355–368 (cit. on p. 94).

[63] Reham Mohamed and Moustafa Youssef. “HeartSense: Ubiquitous Accurate Multi-Modal Fusion-based Heart Rate Estimation Using Smartphones”. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1.3 (2017), p. 97 (cit. on p. 111).

[64] Vinod Nair and Geoffrey E Hinton. “Rectified linear units improve restricted boltzmann machines”. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010, pp. 807–814 (cit. on p. 101).

[65] Rajalakshmi Nandakumar, Shyamnath Gollakota, and Nathaniel Watson. “Contactless sleep apnea detection on smartphones”. In: *Proceedings of the 13th annual international conference on mobile systems, applications, and services*. ACM. 2015, pp. 45–57 (cit. on p. 111).

[66] Babatunde A Otulana, Tim Higenbottam, Lilie Ferrari, et al. “The use of home spirometry in detecting acute lung rejection and infection following heart-lung transplantation”. In: *Chest* 97.2 (1990), pp. 353–357 (cit. on p. 38).

[67] Dominic Papineau, Stephen J Mojzsis, and Axel K Schmitt. “Multiple sulfur isotopes from Paleoproterozoic Huronian interglacial sediments and the rise of atmospheric oxygen”. In: *Earth and Planetary Science Letters* 255.1-2 (2007), pp. 188–212 (cit. on p. 8).

[68] H Paramesh. “Epidemiology of asthma in India”. In: *The Indian Journal of Pediatrics* 69.4 (2002), pp. 309–312 (cit. on p. 32).

[69] Giambattista Parascandolo, Toni Heittola, Heikki Huttunen, Tuomas Virtanen, et al. “Convolutional recurrent neural networks for polyphonic sound event detection”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25.6 (2017), pp. 1291–1303 (cit. on p. 114).

[70] Gene R Pesola, Pamela O’Donnell, Gene R Pesola Jr, Vernon M Chinchilli, and Arthur F Saari. “Peak expiratory flow in normals: comparison of the mini Wright versus spirometric predicted peak flows”. In: *Journal of Asthma* 46.8 (2009), pp. 845–848 (cit. on p. 38).

[71] Andrew B Raine, Nauman Aslam, Christopher P Underwood, and Sean Danaher. “Development of an ultrasonic airflow measurement device for ducted air”. In: *Sensors* 15.5 (2015), pp. 10705–10722 (cit. on pp. 77, 113, 117).

[72] *Respiratory System | Interactive Anatomy Guide* (cit. on p. 12).

[73] Frank Rosenblatt. “The perceptron: a probabilistic model for information storage and organization in the brain.” In: *Psychological review* 65.6 (1958), p. 386 (cit. on p. 94).

[74] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. “Learning representations by back-propagating errors”. In: *nature* 323.6088 (1986), p. 533 (cit. on p. 95).

[75] Olga Russakovsky, Jia Deng, Hao Su, et al. “Imagenet large scale visual recognition challenge”. In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252 (cit. on p. 95).

[76] Justin Salamon and Juan Pablo Bello. “Deep convolutional neural networks and data augmentation for environmental sound classification”. In: *IEEE Signal Processing Letters* 24.3 (2017), pp. 279–283 (cit. on pp. 107, 114).

[77] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. “A dataset and taxonomy for urban sound research”. In: *Proceedings of the 22nd ACM international conference on Multimedia*. ACM. 2014, pp. 1041–1044 (cit. on p. 114).

[78] Steven W. Smith. *The Scientist and Engineer’s Guide toDigital Signal Processing* (cit. on p. 64).

[79] American Thoracic Society et al. “Lung function testing: selection of reference values and interpretative strategies”. In: *Am Rev Respir Dis* 144 (1991), pp. 1202–1218 (cit. on p. 47).

[80] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. “Dropout: A simple way to prevent neural networks from overfitting”. In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958 (cit. on p. 99).

[81] Julian W Tang, Andre D Nicolle, Christian A Klettner, et al. “Airflow dynamics of human jets: sneezing and breathing-potential sources of infectious aerosols”. In: *PLoS One* 8.4 (2013), e59970 (cit. on p. 73).

[82] Geoffrey Ingram Taylor. “Statistical theory of turbulence”. In: *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 151.873 (1935), pp. 421–444 (cit. on p. 78).

[83] *The burden of lung disease* (cit. on pp. 31, 32).

[84] Alan Turing. “Computing Machinery and Intelligence. Mind LIX (236): 433–460”. In: *Reprinted as* (1950), pp. 40–66 (cit. on p. 94).

[85] *Tutorial for MEMS microphones* (cit. on p. 63).

[86] Tarun Wadhawan, Ning Situ, Hu Rui, et al. “Implementation of the 7-point checklist for melanoma detection on smart handheld devices”. In: *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE. 2011, pp. 3180–3183 (cit. on p. 111).

[87] Kristoffer T Walker and Michael AH Hedlin. “A review of wind-noise reduction methodologies”. In: *Infrasound monitoring for atmospheric studies*. Springer, 2010, pp. 141–182 (cit. on pp. 77, 113).

[88] Edward J Wang, William Li, Junyi Zhu, Rajneil Rana, and Shwetak N Patel. “Noninvasive hemoglobin measurement using unmodified smartphone camera and white flash”. In: *Engineering in Medicine and Biology Society (EMBC), 2017 39th Annual International Conference of the IEEE*. IEEE. 2017, pp. 2333–2336 (cit. on p. 111).

[89] Edward Jay Wang, Junyi Zhu, Mohit Jain, et al. “Seismo: Blood Pressure Monitoring using Built-in Smartphone Accelerometer and Camera”. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM. 2018, p. 425 (cit. on p. 111).

[90] Lei Wang, Peder C Pedersen, Diane Strong, Bengisu Tulu, and Emmanuel Agu. “Wound image analysis system for diabetics”. In: *Medical Imaging 2013: Image Processing*. Vol. 8669. International Society for Optics and Photonics. 2013, p. 866924 (cit. on p. 111).

[91] Wikipedia contributors. *Intelligence — Wikipedia, The Free Encyclopedia*. `https://en.wikipedia.org/w/index.php?title=Intelligence&oldid=842159571`. [Online; accessed 25-May-2018]. 2018 (cit. on p. 92).

[92] Wenyao Xu, Ming-Chun Huang, Jason J Liu, et al. "mCOPD: mobile phone based lung function diagnosis and exercise system for COPD". In: *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments*. ACM. 2013, p. 45 (cit. on p. 112).

[93] Yong Xu, Qiuqiang Kong, Wenwu Wang, and Mark D Plumbley. "Large-scale weakly supervised audio classification using gated convolutional neural network". In: *arXiv preprint arXiv:1710.00343* (2017) (cit. on pp. 114, 142, 147).

[94] Jina Yoon. *Comparing 10 Sleep Trackers*. 2017 (cit. on p. 41).

[95] Fatma Zubaydi, Assim Sagahyroon, Fadi Aloul, and Hasan Mir. "MobSpiro: Mobile based spirometry for detecting COPD". In: *Computing and Communication Workshop and Conference (CCWC), 2017 IEEE 7th Annual*. IEEE. 2017, pp. 1–4 (cit. on p. 112).

List of Figures

List of Tables