Investigation of Convolutional Neural Network Architectures for Image-based Feature Learning and Classification

Johnny Ren

A thesis submitted in fulfillment of the
requirements for the degree of

Master of Science in Electrical Engineering

University of Washington Bothell

2016

Committee:

Sohini Roychowdhury, Chair

Mahmoud Ghofrani

Tadesse Ghirmai

Program Authorized to Award Degree:
Engineering and Mathematics

Abstract

Convolutional Neural Networks (CNN) are useful methods for identification of previously unknown embedded patterns in images. Several object and facial recognition along with image segmentation tasks have benefited from the non-linear abstraction of hybrid features using CNN. This work presents a novel CNN model parametrization work-flow developed on the cloud-computing platform of Microsoft Azure Machine Learning Studio (MAMLS) that is capable of learning from the feature maps and classifying multi-modal images with different variabilities using one common flow. This two-step work-flow trains CNN models by splitting the data into training and testing data sets. First, the CNN layers are fixed and the best kernel and normalization parameters that maximize classification accuracy on the test data are identified using grid search. Next, using the best kernel and normalization parameters, the best CNN architecture that maximizes classification accuracy is detected. Finally, the activated feature maps (AFMs) from the parameterized CNN model so far are analyzed to learn new features that can enhance image-based classification accuracies. The proposed flow achieves classification accuracies in the range of 92.5-99.2% that can be further enhanced by doubling the samples based on the features learned from the AFMs. The proposed non-deep CNN models in the MAMLS platform are capable of processing image data sets with 400-4 million samples using a common flow without exponential increase in the computation time. Thus, parametrized non-deep CNN models using the proposed method are capable of identifying novel features that may enhance image-based classification accuracies.

For computed tomography (CT) images, quantitative image assessment can allow for benchmarking image processing methods and optimization of image acquisition parameters. Large volumes of CT images from phantoms and patients are analyzed using the CNN models compare to a baseline model that vary in their implementation time complexities. The

goal here is to model the data set variability for prediction of CT image quality (CTIQ). We observe that for 70% of data samples in training and 30% data sample in test set, respectively, the average multi-class classification accuracies for CTIQ prediction vary significantly as the data sets are switched from the phantom to patient images. The CNN model is found to be more suitable for CT image texture classification in the absence of structural variabilities. Our analysis demonstrates that CNN models are consistent identifiers of structural similarities for CT image data sets. Future work on multi-objective CNN modeling and 3D CNN modeling may lead to new insights for classification tasks.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

Recent advances in computer vision have enabled computers to understand and visualize real-world scenarios. For instance, automated recognition of hand written bank account numbers, checks, and zip codes have significantly reduced manual labor in banking companies for transaction processes [1]. Another example is facial detection technology that is implemented in latest image and video recorders to better focus on the object to be captured irrespective of the incident light. Also, automated facial and expression recognition has led to detection of gender, age, and identities that have been extensively useful to social media, networks, and personalized experiences [2]. Further, applications such as medical image classification and segmentation apply computer vision for automated screening purposes, thereby saving million of dollars in health care costs. For example, the 2015 Kaggle contest on automated Diabetic Retinopathy detection [3] showed that the best entry that applied deep learning strategy secure a score of 0.85 on a (0-1) scale, which is significantly close to human scoring performance. Here, the automated annotation score was measured by the difference levels between human scores and computer predicted scores, that were applied to thousands of retinal images. Besides, three dimensional(3D) object recognition has become increasingly popular because of applications such as robotic arms in automated production systems, augmented reality, and in self-driving cars [4]. Computer vision has countless future possibilities in the field of automated image classification. One of the open challenges for researchers in the field of image processing and machine learning is to develop optimal algorithms and models with respect to recognition performances in order to address such real-world applications.

In this work, we investigate one category of machine learning algorithms useful for solv-

ing computer vision based problems known as Convolutional Neural Network(CNN). We analyze the features of CNN models in detail by accessing the following: the structural and mathematical modeling, embedded patterns, and performing sensitivity analyses of system parameters and architectures variabilities. The following chapters address these concepts.

1. Literature Review: Chapter 2 reviews existing literature on CNN models for 2D image and 3D object recognition. For instance, smart robot systems require automated detection of 3D objects that incorporate not only pixels on the plane but also depth information of the object for classification purposes. Also, several recent works have been directed to digitize 3D objects in several forms of representation.

2. CNN optimal parameterization: Chapter 3 introduces CNN model setup, architecture variabilities, and parameter sensitivities as important factors that affect classification accuracies.

3. Computed Tomography (CT) image quality classification: Chapter 4 presents the importance of automated quantification of diagnostic quality CT images to ensure As low as Reasonably Achievable (ALARA) contrast enhancement dosages [5]. This work is significant because at a low image acquisition tube current, the CT image is noisy or low quality and less noisy or high quality for high tube current, respectively. Thus, this chapter presents our design of a CNN flow to predict CT image quality based on certain combination of pixels in each image.

The organization of chapters in this thesis is as follows. Chapter 2 reviews several recent CNN-based designs for image and object classification tasks to better understand the impacts of system parametrization. Chapter 3 demonstrates a work flow for optimizing CNN parameters and architectures. The proposed three-step flow of parameter optimization, architecture selection, and feature analysis is able to show improved classification results and applicability to three different data sets including one medical data set. Chapter 4 uses real

patient CT image data for CT image classification tasks the methodology presented in this chapter is an extension of the analyses from chapter 3. Finally, conclusions and future works directions are summarized in chapter 5.

1.1 Method and Background

Neural networks are inspired by the neural system in the brain. A multi-layer perceptron consists of layers with several neurons in each layer [6]. The connections in between layers are attached with different weight values. The neurons in each layer respond to different combinations of neurons from the previous layer. A forward prediction (weighted sum) from the previous layer is used to predict the value of the current neuron as shown in Fig. 1.1. Here, an activation function, usually a hyperbolic tangent or a sigmoid, is applied to check if the neuron is triggered. The final output is the sum of all the neuron outcomes multiplied by the weights. The predicted final output is used to compute an error by subtracting the output from the desired class label. This learning process is iterated using the training data samples and is called supervised learning. The error per target value is then used to compute the gradient of error with respect to activation function and the forward prediction function. This process is called back propagation. At the end of the back propagation process, all the weights and biases in the neural system will be updated in order to predict probabilities for class labels of test sample with minimum error. Finally the predicted probabilities are thresholded to obtain test sample class labels as shown in Fig. 1.1.

CNN models are an extension to neural networks where regional features are extracted and trained for classification purposes. Two important features that drive CNN models are the convolutional layer and sub-sampling layer. CNNs mimic our visual cortex system, where the local receptive field is key to identification of image patterns. In the convolutional layers, several two dimensional kernels convolve with the input image pixel by pixel from top left to bottom right like a 'sliding window' [7]. While functioning as local receptive fields in our eyes, elementary features are extracted in the first few layers such as edges extraction from various orientations. The sub-sampling layer performs max/average pooling

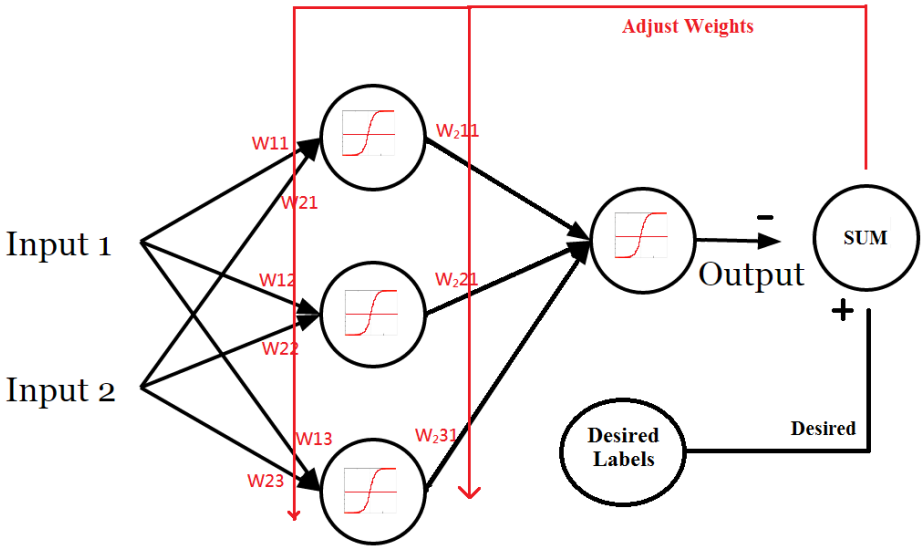

Figure 1.1: Illustration of a feed-forward neural network and back propagation of the error.

thereby enhancing features from the previous layer. Next, activation function is applied on the sub-sampled images to filter and obtain the weighted sum of pixel intensity with the kernel weights and bias as shown in Fig. 1.2. Finally a fully connected neural network is used for the classification task.

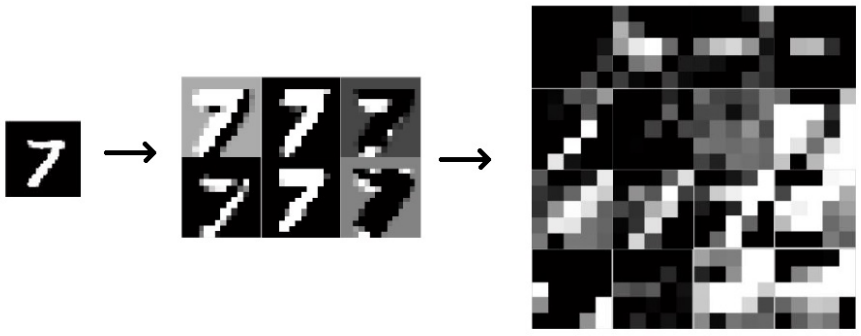

Figure 1.2: Feature extracted for a handwritten digit '7'.

There are three reasons for the choice of CNN model in this investigative research.

1. The CNN model saves on memory space and computational time when compared to traditional linear models for analysis of highly pixelated images. For example, the MNIST handwritten digit recognition model by Yann Lecun [7] had only 3550 parameters changing overtime when compared to neural network model for an image with [784x784] pixels, which represents 614656 parameters. Thus, CNN is capable of learning such high volumes of model parameters relatively fast.

2. For a traditional neural network, vectorized images lose their spatial relationship with neighboring pixels. CNN extracted spatially significant pixels thereby overcoming the over-fitting limitations of neural network models.

3. Some variabilities introduced by imaging systems include variations in fields of view, orientations, vertical and horizontal shifts. The CNN model is shift invariant because the same learned kernels parameters are applied across the whole image space. The strong activation triggers whenever it detects a pattern [8].

Thus, the CNN model applies the concepts of local receptive fields, shared kernel weights, and sub-sampling for robust, reliable feature extraction from multi-class image-based large data sets.

1.2 Material and Methods

In this work, we use a toolbox that is available on MathWorks to build a CNN structure for all classification tasks [9]. For computational convenience, we utilize the Microsoft Azure Machine Learning Studio Platform (MAMLS) [10] to set up multiple architectures for medium to large scale CNN data modeling for classification tasks. We use MATLAB to observe AFMs from the hidden layers. The objective of this research is to analyze the communication of information and underlying pattern among the hidden layers for the classification tasks.

Chapter 2
LITERATURE REVIEW

Over the past decade CNN models have been widely used for automated image classification, image tagging and cognition-based tasks [11]. Deep learning models became a preferable solution to Big data processing problems with the access to supercomputing, distributed computing and cloud-computing frameworks. As the high-speed computation and capabilities and hardware independence increased, the limitations posed by baseline NN models regarding memory storage and weight updations significantly diminished. This led to CNN model parameter tuning and information retrieval projects to be developed on "Big data" sets that needed several GB to TB of storage space. The goal being investigative research for previously unknown patterns or trends in the data. Some existing state-of-the-art works for CNN implemented on big data sets, medical image data set and 3D object recognition data sets are summarized below.

2.1 Big data

One prominent work on CNN-based analysis of images in [12] presented an "adversarial example" that was analyzed specifically on the MNIST handwritten digit recognition data set [1]. This work focused on addition of perturbations to images to identify the limitations in image classifications with varying degrees of added perturbations. In this work, we extend this idea of adding variabilities, such as image edge thickening and image flipping, to the input image data sets to assess the importance of the AFMs learned from a trained CNN model. Another work in [13], implemented de-convolutional networks to visualize the strongest feature map activations on 4 major image-based big data sets including ImageNet. This method tracked the activated kernels to identify the activated feature filters that aided

feature classification. Also, an occlusion sensitivity analysis was performed to show the relationship between occluded spatial locations and classification probability scores. In this work, we extend our analysis to AFMs based on domain knowledge to identify new embedded features for specialized multi-modal image-based data sets. The other method that followed ImageNet competition for image classification [14] presented a method to boost image classification accuracy by super-sampling the input images. This method super-sampled each input image by 4 corner-based crops and 1 image center-based crop followed by each super-sampled image flipping. Thus, 10 new images were generated out of each input image. In this work, we explore the input image super-sampling strategy for its generalizability across multi-modal image data sets.

One of the recent methods that analyzes CNNs in bio-medical applications [15], presented the importance and effects of model parameters in computational neuro-science to emulate the sensory cortex. This method presented that deep CNNs often fail in medical image-based research settings due to the lack of input data variabilities. In such scenarios, the deeper layers do not learn any new features and hence non-deep layers are preferable for reasonable feature learning. This work also estimated that for optimal CNN training, the number of input samples must be at least 10 times the number of network parameters to be trained. Thus, non-deep CNN models are preferable in the absence of high numbers of input samples and considerably low image variabilities. This work utilizes non-deep CNNs for analysis of AFMs in input image data sets that have low to moderate degrees of variabilities for optimal classification of images.

2.2 Medical image: CT image quality

Artificial Neural Networks (ANN) have been applied to extract features for the identification of diseases, such as those in the lung as shown in [16]. In this work, the first step involves the segmentation and selection of the lungs by thresholding. The segmented lungs are further processed to extract the relevant features used for disease identification, which in this case is the auto-correlation of the pixels in the lung. The ANN classifier is shown to identify various

lung diseases by this method. However, the results of the auto-correlation are impacted by the noise and quality of the CT image, thereby making the ANN classifier dependent on the CT imaging parameters and the resulting CTIQ. In addition, the choice of features used in the ANN classifier may not be optimal for every patient. Thus, the development of quantifiable metrics for CTIQ, as well as the use of a CNN model to extract relevant features, can lead to generalizable data modeling.

Finally, the work in [17] demonstrates the application of machine learning to suppress image artifacts and noise in low-dose CT images, specifically by dictionary learning. Distinguishing artifacts from actual features in images is a traditionally challenging problem that is addressed in this work by expanding the dictionary to include artifact atoms trained along with the standard feature atoms. This discriminative dictionary can be used to identify both standard features and artifacts separately, allowing the latter to be suppressed. This method was tested with abdominal and mediastinum CT images and compared to the CTIQ scores provided by radiologists. However, the process of learning these discriminative dictionaries requires parameters and ROIs that are empirically set. Thus, this method is applicable only to low-dose CT images with similar image acquisition parameters. Additionally, the discriminative dictionaries can significantly suppress artifacts, but random noise remains continues to impact the performance of the sparse coding model. The need for regularization terms in this method can be addressed through automated ROI selection methods and quantifiable CTIQ metrics, such as those discussed in this paper.

2.3 3D Object Recognition

Followed by the success of CNNs on 2D image classification tasks, researchers have extended CNN applications to 3D object recognition. Several methods have been developed to address the technical limitations of present day technology. Studies have shown that there are 2 primary methods for a good representation of 3D objects data [4][18][19][20][21]. First, a 2.5D representation measures the depth information instead of intensities of pixels in a 2D image. One challenge of 2.5D object recognition is shape completion. Here, the structure

behind the point of view is blocked, therefore the underlying features and patterns can not be estimated from a single view. This leads to over-fitting instance recognition and classification tasks. Second, methods such as [4] introduce 3D volume representations that convert point cloud data, which is collected through RGB-D camera or LiDar [4][18] to a 3D binary occupancy grid space. Some of the existing works that apply these two category of methods are described below.

In [21], the work multi-views approach is introduced that augments the data to several different orientations. The work in [18] uses pre-trained 'Alexnet' model[22] to further train with multi-views of objects. Collections of several different directions of 2D projected images pass through an image-based CNN architecture for classification. The works [21][18] combine and vote from classification results of all orientations to further perform object reconstruction and recognition. Although these methods well utilize depth information, they lack critical descriptive information such as color and texture of the object. Although, the 3D CNN models it is also lack color and texture information, they fully exploit the spatial information to achieve optimal classification accuracies. In [4][18][19][20], data augmentation is performed in order to maximize the intra-class classification accuracies as well as minimize the inter-class variation. The second category of methods involving 3D object recognition are applied to multi-views thereby allowing rotating and shifting in 3D spaces. Among the primary works on 3D CNN modeling [4], CNN apply 3D cnn kernels as opposed to 2D kernel that were previously applied to 2D images. These 3D kernels appear in box like shapes representative of the volumetric voxels.

Chapter 3

CNN MODEL PARAMETRIZATION FOR MULTI-MODAL IMAGE CLASSIFICATION

3.1 Introduction

Big data analytics provide the apparatus for learning unforeseen patterns and trends in large sets of data that can lead to predictive modeling for a variety of applications. Some well-known examples include automated hand written digit recognition that has significantly benefited the postal delivery system, and automated facial recognition that has significantly improved security and privacy protocols. Also, medical image quality detection and regulation function can significantly improve the present day health care resourcefulness [5]. This work is aimed at improving the performances of predictive models by parameterization using a generic work-flow that is robust to variations in input data streams and initial conditions. Such a system will be useful for a wide variety of image-based predictive modeling tasks such as medical image-based detection systems for personalized medicine, automated handwriting recognition and facial identification systems.

Over the past decade, deep learning strategies have been used for a wide range of machine learning and pattern recognition tasks [23]. Deep learning refers to extracting higher-orders of embedded features by non-linearly processing several layers of low-order features. This process uncovers previously unknown trends or patterns in the combination of features, which can aid predictive modeling from data. For instance, several non-linear combinations of pixels from an image may be learned as the embedded characteristics required for image-based classification tasks. With growing sizes of image-based data sets, deep learning strategies such as the Convolutional Neural Networks (CNN) [24] have emerged as key providers for embedded patterns in images when compared to traditional machine learning algorithms [9].

Over the last few decades and with the advent of high-speed computing solutions, the use of CNNs has expanded to "Big Data" sets that involve pattern recognition from millions of sample images. The 2015 Kaggle contest on automated Diabetic Retinopathy (DR) detection demonstrated that CNN-based approaches are capable of detecting the DR lesions that are manifestations of the retinal pathology [3]. Another well-known application of CNN is the ImageNet competition that demonstrates the scalability of CNN to categorize over 1000 different classes of images with approximately 90 percent accuracy on validation data set using 1.2 million images for training purposes [22].

One limitation of deep learning models on relatively smaller data sets, where the number of samples is significantly lesser than the data dimension, is "over-fitting". In such instances, the trained model lacks generalizability and may result in high classification errors on previously unseen data sets. In this work, we present a common CNN work-flow implemented on the cloud-computing platform of Microsoft Azure Machine Learning Studio (MAMLS), which is capable of automated parametrization for object and pattern recognition tasks on a variety of image-based data sets. Our model trains a sequence of feature maps and feature weights using gradient-based back-propagation on CNN. However, the overall classification accuracy is dependent on the architecture of the CNN model, i.e., the order and number of deep layers in the model [22]. The goal of this work is to determine the best architecture and parameters that can be automatically tuned to learn new hierarchical features and perform automated classification tasks irrespective of the input image source.

Prior to the popularity of CNN, image-processing using baseline neural network (NN) architectures was implemented using several layers of neurons with thousands of neurons per layer that were partially or fully connected. In such NN models each input layer neuron corresponded to every pixel intensity in the image [25]. The major drawback of this setup was that for highly pixelated images, the number of weights and biases, corresponding to neuron connections in a fully connected network that had to be iteratively updated, imposed heavy memory storage requirements. Also, the spatial inter-dependencies in the image were lost in such a set-up. Additionally, neural networks were invariant to spatial, configural and

resolution variations in the input images. Thus, slight spatial variations in the training image data affected the trained parameters of the network, thereby missing the structural features embedded in groups of pixels. CNNs pose viable solutions to all these limitations of baseline NN architecture, such that non-linear combinations of sub-sampled spatial configurations can be utilized to learn hybrid features while maintaining the spatial configurations of each input image.

Over the past few years CNN have been utilized in numerous deep learning tasks owing to its multi-layered hierarchical architecture with inter-separated convolutional and sub-sampling layers that capture spatial and configural features [7]. Each CNN layer generates a different response to the non-linear combination of inputs from the previous layers. While the first layer generally detects elementary features such as edges in different orientations, higher-order combinations across sub-sampled spatial orientations can be detected in the deeper layers [8]. MAMLS provides a hardware independent platform that is capable of performing several thousands of parameter tuning operations on input data sets with storage sizes up to 20 GB [10]. This work applies the MAMLS-based CNN models as Big Data analytics tools to learn new unforeseen features that can enhance classification tasks for images that are acquired from a wide variety of cameras and modalities.

This paper makes three key contributions. First, a 2-step process is presented that is capable of learning the best CNN model parameters and architecture for a particular data set using grid search method. The performances of these settings are comparatively analyzed with respect to state-of-the-art existing works. Second, non-deep CNN architectures are found to extract new features from activated feature maps (AFMs) of multi-modal sets of image data acquired for digit, face recognition and CT image quality detection. Such multi-modal assessment of images has not been done on a single work-flow so far. Third, the proposed CNN work-flow in the MAMLS platform is capable of classification of data sets with storage sizes 12 MB-10.1 GB in one hardware independent generic work-flow without significant increase in the computation time. An example of the proposed flow is presented in Fig. 3.1.

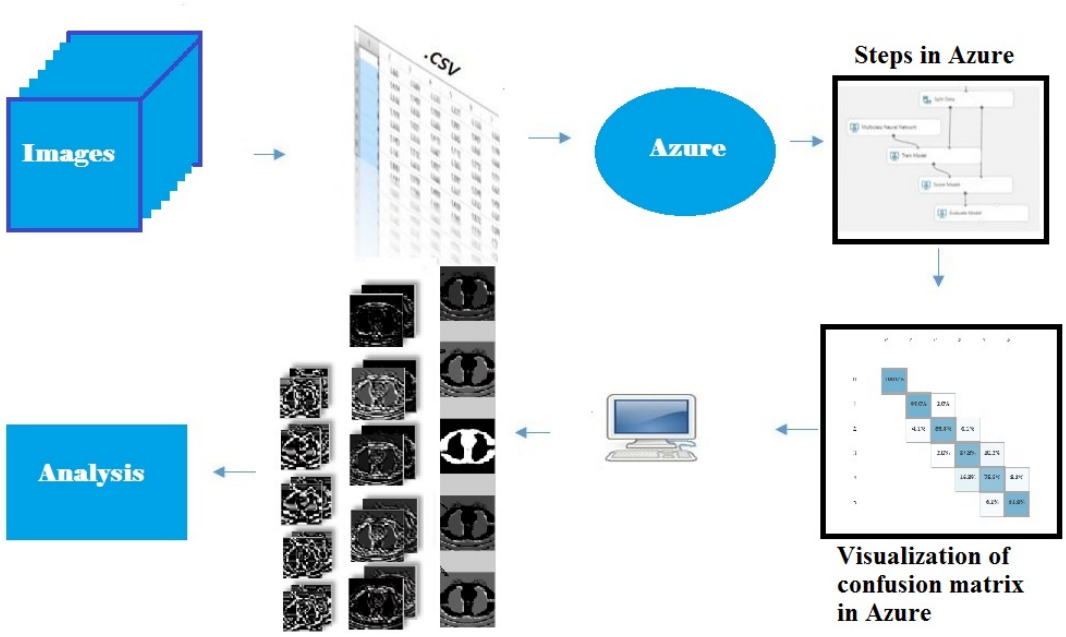

Figure 3.1: The proposed CNN modeling work-flow using MAMLS platform. Each image is converted to a sample row in a spreadsheet-like flat-file format. The spreadsheet-like data is then uploaded to the cloud-platform, followed by CNN modeling, classification and visualization in the MAMLS environment. Finally, the AFMs corresponding to the deep layers are analyzed to learn new features from the activated regions.

3.2 Materials and Methods

CNN models can be parameterized for few to several deep layers for extracting embedded features from groups of image pixels. Hence, it is imperative to identify the best CNN architecture that is capable of extracting embedded features in image data sets. In this work, the best CNN architecture is first identified in the MAMLS platform followed by analysis of the AFMs from the deep layers for learning new features. For an image data set with n images, each input image is denoted as $X_k, k \in [1:n]$ with [rxc] pixels with the output sample class label $Y_k = t : t \in [1, 2..T]$, where T represents the number of target output categories. Each 2-D image is resized as a single row with its corresponding output and transferred to the MAMLS platform in flat-file formats of 'csv', 'libsvm', 'txt' etc. The

CNN model for each data set is trained by stratified sampling using 70% of the data samples and tested on the remaining 30% samples (also known as the 70/30 data split). Stratified sampling ensures similar sample class frequency distributions in the training and test data sets [26]. The image data sets, the CNN model parameters and architecture are explained in the following subsections.

3.2.1 Data

Three modalities of data sets are analyzed for image-based classification tasks using CNN. These data sets represent varying degrees of input image variabilities. The first data set with low spatial variability comprises of abdominal computed tomography (CT) images acquired for a phantom by varying the image acquisition parameters. The second data set with mild spatial variability comprises of a set of facial close-up images that can be used for facial identification tasks. The third data set with moderate spatial variability is a set of handwritten digits corresponding to variable writing styles and spatial structures. These data sets are explained as follows:

1. CT Image Data Set (Low variability): Using the Lightspeed 16-slice scanner where most system settings were kept constant (helical, pitch 1.375,20 mm collimation, 120 kV p), a dataset of abdominal phantom CT images is acquired by varying the tube current values of 10, 25, 75, 125, 175, and 350 mA, respectively. Corresponding to each tube current setting, 6 sets of 81 abdominal CT images are acquired (486 images). Since tube current values are directly proportional to the CT image qualities, the CT image quality varies as $Y_k = [0, 1, 2, 3, 4, 5]$ and $T = 6$ target categories as the tube current is varied from 10 ma through 350 mA, respectively. Here, $Y_k = 0$ refers to a noisy abdominal CT image while $Y_k = 5$ is very high quality image. Each CT image has dimensions [512x512] with pixel intensities in the range [-5000,5000]. This locally acquired data set has a storage size of 1.22 GB. The images from this data set are pre-processed to [119x119] by sub-sampling 4 times and discarding the image edges

along with pixel intensities in [0,1]. The stratified sampling for this data set is shown in Fig. 3.2.

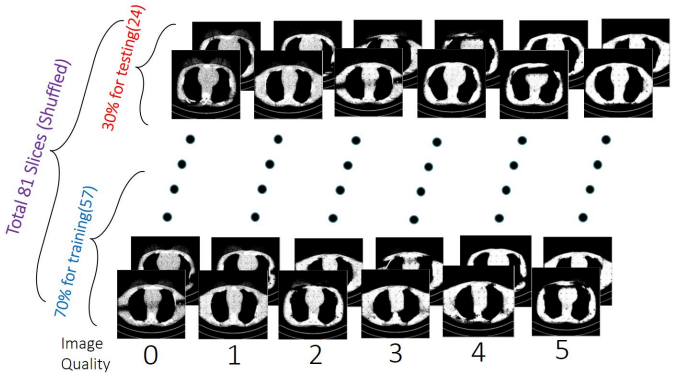

Figure 3.2: The proposed 70/30 data split by stratified sampling for the local CT image data set.

2. Face data set (The ORL Database of Faces [27], Mild variability): This public data set contains 400 facial close-up images with varying spatial configurations obtained from 40 distinct subjects ($T = 40$ target categories) with 10 images per subject. Each facial image has varying degrees of lighting, posture (facing left, right or straight), facial expression and facial details (glasses / no glasses, beard, mustache etc.). All input images are centrally positioned and normalized for pixel intensities in range [0,1]. This data set with each input image of size [92x112] pixels has a storage size of 12 MB.

3. MNIST (Handwritten digit recognition [7], Moderate variability): This public dataset contains samples from a variety of hand writing styles for digits 0 through 9 ($T =$ 10 target categories). Each image is centered and of size [28x28] pixels with pixel intensities in range [0,1]. Two versions of this data set are available. In the first smaller data set (MNIST small) of storage size 125MB, there are 60,000 image samples for training and 10,000 samples for testing with supervised labels for each sample. In the second larger data set (MNIST big) of storage size 10.1GB, there are 4 million digit image samples with supervised labels each.

For each image, the pixel intensities are initially normalized in [0,1] using the min-max normalization strategy.

3.2.2 The CNN Model

The CNN architecture used for data modeling is motivated by the method in [7]. An input image to a CNN model is processed through several deep feature learning layers to generate an output decision (prediction) vector whose length is equal to the number of classification categories. There are four kinds of layers in a typical CNN architecture: convolutional (C), sub-sampling (S), activation and normalization (AN) and dense convolution (D). Each convolutional layer transforms a set of feature maps into another set of feature maps by convolution with a set of kernels/filters. The numbers of each kind of layer and the order in which they are laid out vary across architectures. In this work, we represent ϕ as the CNN architecture that defines the numbers and ordering of the layers. For an input data set, where each input image $X_k, k \in [1:n]$ is classified to a target category $Y_k = t : t \in [1:T]$ or $t \in [0:T-1]$, the parameters for the CNN model are defined as follows.

The interactions between the model layers are analyzed such that, if $W_i^{l_\phi}$ and $b_i^{l_\phi}$ represent the weights and biases of the i'th filter/kernel of the l'th convolutional layer, then $H_i^{l_\phi}$ represents its feature map in (3.1). The kernels of each layer have dimensions $[N^l \text{x } U^l \text{ x } V^l]$, where N^l, U^l, V^l represent the number, width and height of the N^l2-D kernels, respectively. These kernels are initialized to random values and they are trained by back-propagation using the training sample set. The convolution of these kernels with the input images result in the feature maps for the entire layer (H^{l_ϕ}). At the end of the convolution layer (C), the resulting images are sub-sampled/pooled followed by the application of an activation function such as the 'tansig' function in (3.2). For all pixels in image X_k with intensity 'm', the activation function rescales each sub image 'i'' in range [-1,1]. Each resulting image is then normalized using Gaussian or min-max normalizations (AN layer). At the end of all C-S layers, defined by architecture parameter ϕ, each resulting feature map is convolved with a dense set of 3-D kernels (D layer) that are connected to a hidden layer of NN models (NN

layer with 200 neurons). The final output of the NN model is a vector of variables $H_{j,out}(k)$ for image number k, where $j \in T$. Here, the neuron with the maximum value is chosen as the target class in (3.3). The squared error in classification ($Err(k)$) is back-propagated to train the feature maps in each CNN layer (3.4). Finally, different orientations of non-deep architectures (ϕ) are analyzed to select the best CNN architecture (ϕ^*) that minimizes the mean squared classification error in (3.5).

$$H_i^{l_\phi}(k) = H^{l_\phi - 1} \otimes W_i^{l_\phi} + b_i^{l_\phi}. \tag{3.1}$$

$$\forall m \in X_k,\ H_{i'}^{l_\phi}(k) = tansig(m) = \frac{2}{1 + e^{-2m}} - 1. \tag{3.2}$$

$$H_{out}^{\phi}(k) = \arg_j \max H_{j,out}^{\phi}(k). \tag{3.3}$$

$$Err^{\phi}(k) = [Y_k - H_{out}^{\phi}(k)]^2. \tag{3.4}$$

$$\phi^* = \arg_\phi \min \frac{1}{n} \sum_{k=1}^{n} Err^{\phi}(k). \tag{3.5}$$

To train CNN model, $W_i^{l_\phi}$ and $b_i^{l_\phi}$ are initialized randomly followed by back propagation, resulting in the trained feature maps H^{l_ϕ} also known as AMFs. In CNN models, units in the deeper layers indirectly interact with a larger spatial input region, thus forming a high level abstraction of the input data. At the end of model training, the AFMs can be further analyzed for additional feature learning. For the locally generated CT image data set, the architecture with $\phi = [C - S - AN - C - S - AN - D - NN]$ is shown in Fig. 3.3. This architecture contains 7 deep layers before the NN classifier. Other possible architectures/variations in ϕ are presented in the following subsection.

3.2.3 MAMLS Setup

In this work, CNN models are built on the MAMLS platform to leverage its hardware independence and high-speed distributed computing capabilities. The MAMLS platform is user-friendly and it can process up to 20GB of input data size for predictive modeling tasks. The distributed computing feature of MAMLS allows parallel execution of up to 4 modules

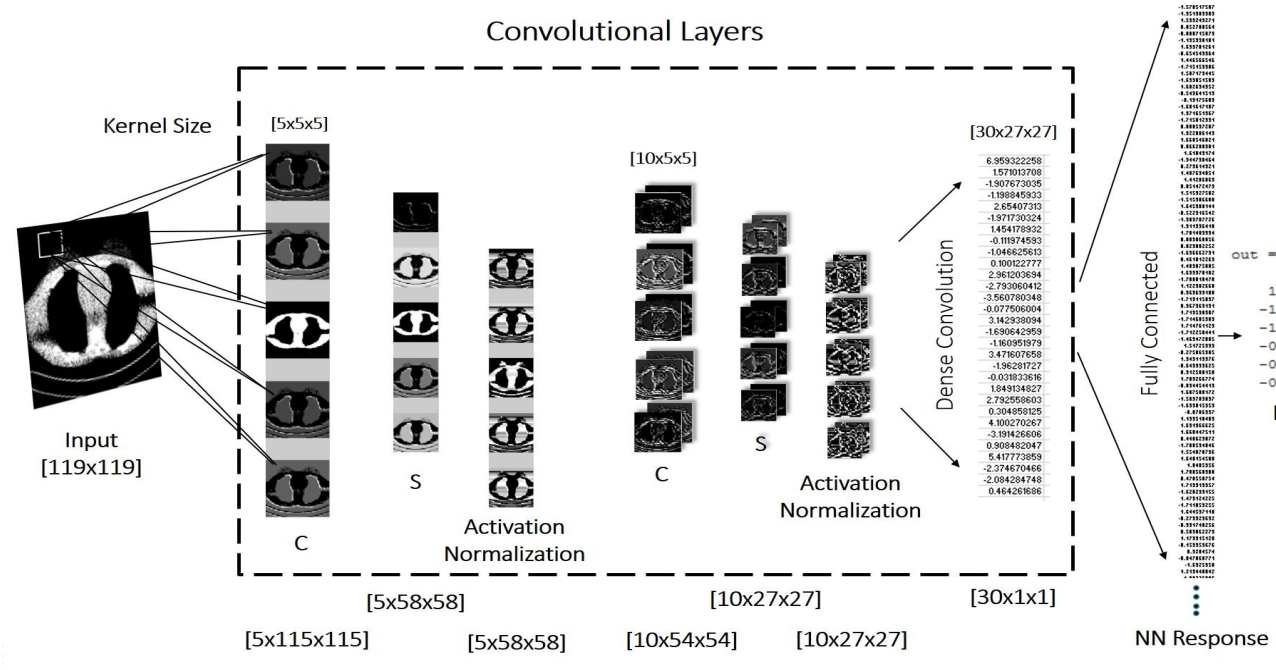

Figure 3.3: The underlying CNN model paremeters using the CS^2 architecture. The kernel sizes and feature maps obtained using one combination of kernel and activation function parameters is demonstrated on the CT image data set.

in one experiment and up to 10 experiments at the same time. The proposed system flow of CNN model in the MAMLS platform is shown in Fig. 3.4. The classification performance of the trained CNN model is evaluated in terms of the number of correctly classified samples (true positives and true negatives) and the number of miss-classifications (false positives and false negatives). For multi-class classification problems, the micro-averaged precision, recall and accuracy are analyzed respectively [10]. The micro-averaged accuracy (ACC_μ) is indicative of the ratio of correct classifications weighted by the sample class frequencies. The best CNN model parameters and architecture are identified in 2 steps. First, for a fixed standard CNN architecture, the best kernel parameters U^{l*}, V^{l*} and normalization function (AN^*) that maximize average classification ACC_μ are computed. Next, for the best kernel and normalization parameters, the CNN architecture (ϕ^*) that maximizes average ACC_μ is identified as follows:

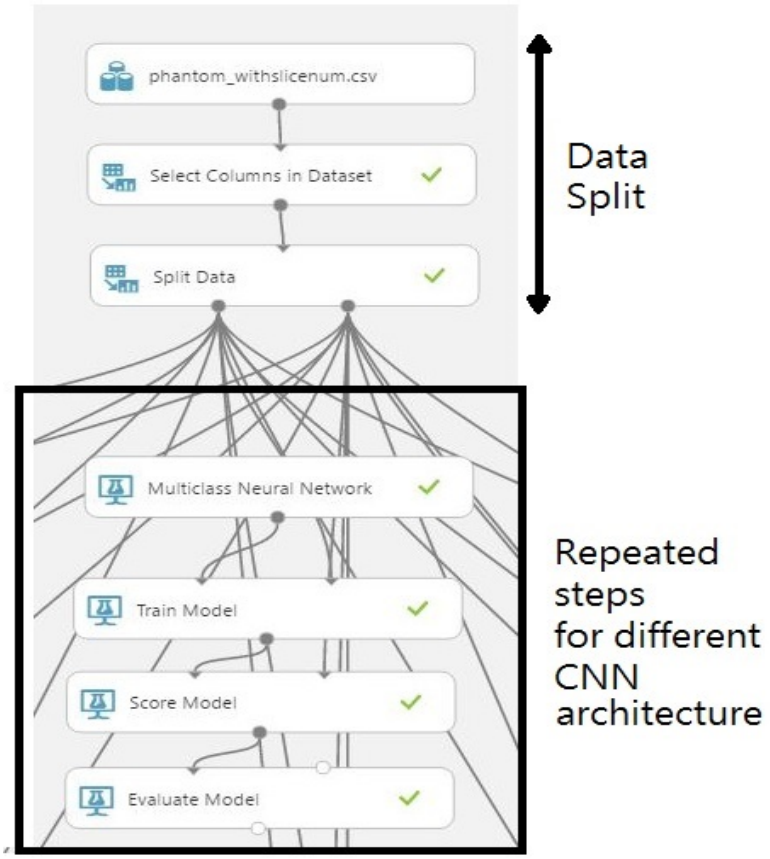

Figure 3.4: The generalized MAMLS setup for CNN model parameterization. This flow automatically parametrizes the model for each individual data set. First, 70/30 data split is applied. Next, CNN module is parameterized on training data, and scored on the test data followed by evaluation of performance metric.

Fixed Architecture, Varying Parameters

In the first step towards CNN model parameterization, the CNN architecture prior to the NN classifier is fixed to [C-S-AN-C-S-AN-D] (7 layers) based of prior works in [7][28]. Next, we vary the kernel size (U^l, V^l) in a range of [3:2:21], while keeping the number of kernels (N^l) fixed as [5, 10, 30] as shown in Fig. 3.3. If the kernel size exceeds the dimension of the resulting feature map, zero-padding is applied to fit the kernel. We observe empirically that the CNN model is more sensitive to the kernel size than the number of kernels. The

other CNN parameter is the normalization method. Motivated by the prior work [13] that demonstrated that normalization can improve accuracy in classification tasks, we varied the normalization function from min-max and Gaussian normalization to identify the best parameter for each data set.

Fixed Parameters, Varying Architecture

In the second step towards CNN model parameterization, using the best U^{l*}, V^{l*}, AN^* identified in the previous step, each data set is subjected to different CNN architectures (ϕ) to identify the architecture that maximizes ACC_μ. These variations in non-deep architectures are defined in Table 3.1. The total number of layers is calculated in Table 3.1 by adding 3 additional layers corresponding to the AN layer, D layer and the classification (NN) layer.

Table 3.1: CNN architectures for best CNN model parameterization.

Architectures		
ϕ	Meaning	Total Layers
C^q	'q' consecutive C layers for $q \in$ [1:3]	q+3
$(CS)^q$	'q' consecutive CS layers for $q \in$ [1:3]	2q+3
$(CCSS)^q$	'q' consecutive CCSS layers for $q \in$ [1:2]	4q+3
$(CCS)^q$	'q' consecutive CCS layers for $q \in$ [1:2]	3q+3
$(CCCS)^q$	'q' consecutive C layer for $q \in$ [1:2]	4q+3

3.2.4 Feature Learning: AFMs

Once the best parameterized CNN model is trained, the AFMs are analyzed [9] to learn new embedded features that impact image-based classification tasks. Based on the activation regions that are fired up in most of the feature maps, two categories of features are qualitatively identified, namely: image orientation and enhanced edges. The image orientation

feature is implemented by flipping each image by 180^o or 90^o, thus doubling the sample size to enhance input sample variability. The enhanced edge feature is implemented by adding the edge information, detected by the 'Sobel' filter, to the original image. These two categories of features learned from the AFMs are shown in Fig. 3.5. The impact of these features are assessed in terms of the variations in ACC_μ caused by doubling the sample size.

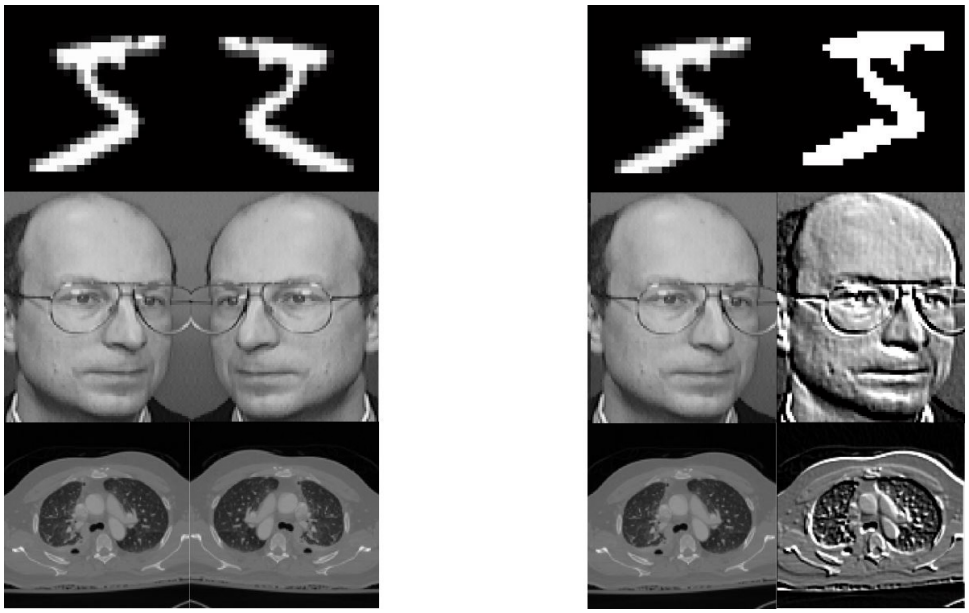

Figure 3.5: Demonstration of modification to input image. The left column shows original image and its flipped version. The right column shows images with enhanced edges.

3.3 Experiments and Results

Four categories of experiments are performed to analyze the contributions of the proposed flow. First, the performance of our method is compared with existing state-of-the-art methods for image-based classification tasks. Second, the performance of the best parameterized CNN models is analyzed for the three multi-modal image data sets. Third, feature analysis is performed to analyze the performance of the proposed CNN models for classification tasks. Forth, the importance of the best CNN model parameterization and sensitivity to training

data is analyzed for a combined set of multi-modal images.

3.3.1 Performance analysis with Existing Works

In Table 3.2, the classification performance of the best parameterized CNN model is analyzed with respect to existing state-of-the-art methods. The baseline method for comparison is a NN model with one input, one hidden layer (200 neurons) and one output layer. The learning rate for CNN and baseline NN models is 0.05. The bench-mark method on the MNIST small data set in [29] performs extreme learning with $\geq$20,000 hidden neurons using the NN classifier and achieves 98.9% classification accuracy on a 3GHz dual-core i7 8GB RAM system in 17 minutes. In comparison, the proposed work-flow achieves 99.1% ACC_{μ} in 22 minutes of cloud-processing time for 10 epochs processed with a learning rate of 0.2. The computation time in MAMLS platform is also affected by queuing delays and server wait times, thus the larger processing times. It is noteworthy the processing time for MNIST big data set is 35 minutes due to the consistent training data set with respect to the MNIST small data set. This is indicative of scalable processing nature of the MAMLS platform as the data sizes change from 104.5 MB to 10.1 GB.

We observe comparable performance on the MNIST big with respect to existing work in [30], that requires several hours for model training. For the Face data set, using 200 hidden neurons in NN model, the method in [25] required 2 days on a 100MHz system for training the classifier to achieve 87-90% classification accuracy with 50/50 data split. For the CT image data set, the image segmentation method in [5] achieved only 75% classification accuracy using one segmented feature metric. Thus, in such medical image data sets, CNN models are useful for underlying pattern recognition.

From Table 3.2 we observe that the proposed CNN models demonstrates almost similar to improved ACC_{μ} when compared to existing works for the three modality of input image data sets.

Table 3.2: The performance of proposed flow with respect to existing works

Data set	Baseline	Existing	Proposed
	$ACC_{\mu}(\%)$		
MNIST small	97.14	[29] 98.96	**99.2**
MNIST big	93.15	[30] **99.3**	98.9
Face	82.50	[25]87-90	**92.5**
CT Image	75.60	[5]75	**93.7**

3.3.2 Performance Analysis

The importance of the 2-step CNN modeling parameterization method proposed in this work is analyzed. In the first step, for a fixed CNN architecture, the impact of variation in kernel size and normalization function on the ACC_{μ} for all the three data sets is shown in Table 3.3. Here, we observe that for the MNIST small data set, best kernel size is [5X5] with min-max normalization function. For the relatively larger images in Face and CT image data set, the best kernel size is [9x9] with min-max and Gaussian normalization functions, respectively. The number of epochs for back-propagation is kept constant to 100.

The second step of CNN model parameterization modifies the CNN architecture (ϕ) using the best selected model parameters from Table 3.3. Here, all hidden layer images are maintained at the same size for each parameter setting in 3.3. For each data set, the classification performance by varying CNN architectures is shown in Table 3.4. Here, we observe that for the MNIST, Face and CT Image data sets $\phi^* = [C-C-S-S]$, $\phi^* = [C-S]$, and $\phi^* = [C-S-C-S]$, respectively. It is noteworthy that too-shallow (3 layers) and too-deep (9-11 layers) architectures result in high classification errors.

For the best parameterized CNN model (with C-S-C-S-D-NN) layers on the CT image data set, the number of parameters to be trained are: kernels weights and biases in C^1 layer (5x5x5x1+5), S^1 layer (5+5), kernel weights and biases in C^2 layer (10x5x5x5+10),

Table 3.3: Sensitivity analysis of fixed CNN architecture to variation in kernel parameters and normalization function

7 Layers CNN							
	MNIST small		Face		CT Image		
	Gaussian	Min-Max	Gaussian	Min-Max	Gaussian	Min-Max	
Filter Size	$ACC_\mu(\%)$						Sub-sampling Stride
3x3	98.3	98.4	62.5	3.00	78.4	42.4	[2x2],[2x2]
5x5	**98.8**	98.7	85.8	18.3	83.3	44.4	[2x2],[2x2]
7x7	98.3	98.6	77.5	75	90.3	65.9	[2x2],[2x2]
9x9	98.2	98.6	76.6	**91.7**	**93.7**	59.0	[2x2],[2x2]
11x11	98.1	98.6	80.0	86.67	89.5	53.4	[2x2],[2x2]
13x13	97.8	98.5	73.3	88.3	87.5	73.6	[2x2],[2x2]
15x15	98.1	98.6	70.8	89.2	86.8	71.5	[2x2],[2x2]
17x17	98.0	98.6	72.5	91.7	86.1	70.8	[2x2],[2x2]
19x19	97.8	98.6	74.2	88.3	79.8	63.8	[2x2],[2x2]
21x21	97.8	98.4	71.6	86.67	82.6	66.7	[2x2],[2x2]

S^2 layer (10+10), dense convolutions in D layer (27x27x30x10+30), hidden layer for NN model (200x30+200) and output layer for NN model (200x6+6). Thus, for this data set 118,206 units of parameters have to be estimated. Since the CT image data set has only 486 samples, the trained CNN model suffers from over-fitting errors, thereby resulting in low $ACC_\mu = 93.7\%$. This observation is inline with existing medical image based methods that are over-fitted due to lack of sufficient medical data [15]. The best parameterized CNN model performance on this local data set is shown in Fig. 3.6. The proposed CNN models result in highest ACC_μ for the MNIST data set, since the images in this data set have moderate variability with a large number of training samples, thus reducing the chances for over-fitting. However, for the Face data set, that has relatively lower variability and number of samples

Table 3.4: Variation in Classification Performance with variable CNN architectures. ACC_μ is in %. Time is in minutes.

Data		MNIST small		Face		CT Image	
Φ	#Layer	ACC_μ	Time	ACC_μ	Time	ACC_μ	Time
C^1	4	98.1	52	87.5	6	67.4	16
C^2	5	98.4	1028	2.50	101	33.3	257
C^3	6	98.5	2438	2.50	215	16.6	1431
CS^1	5	98.6	26	**92.5**	**3**	84.0	5
CS^2	7	98.8	108	90.8	19	**93.7**	**26**
CS^3	9	98.9	216	2.50	34	65.2	52
$CCSS^1$	7	**99.2**	**309**	80.8	37	87.5	79
$CCSS^2$	11	98.7	535	2.50	176	76.4	105
CCS^1	6	98.8	388	2.50	52	77.8	105
CCS^2	9	98.8	1742	2.50	190	16.6	748
$CCCS^1$	7	98.7	2846	2.50	274	16.6	543

when compared to the MNIST data set, the proposed CNN models result in some degree of over-fitting that results in the lowest ACC_μ across all the three data sets. Fig. 3.7 shows the examples where faces were miss-classified. We observe that since CNN primarily detects shapes, the miss-classified faces have very similar facial shapes and orientations. However, for the borderline classification cases, although the faces are correctly recognized, the closest matched faces have similar facial structures and fields of view.

3.3.3 Feature Learning from AFMs

The MNIST small data set has been used extensively to bench-mark performances of machine learning algorithms. Due to the large sample size of this data set, the classification accuracies

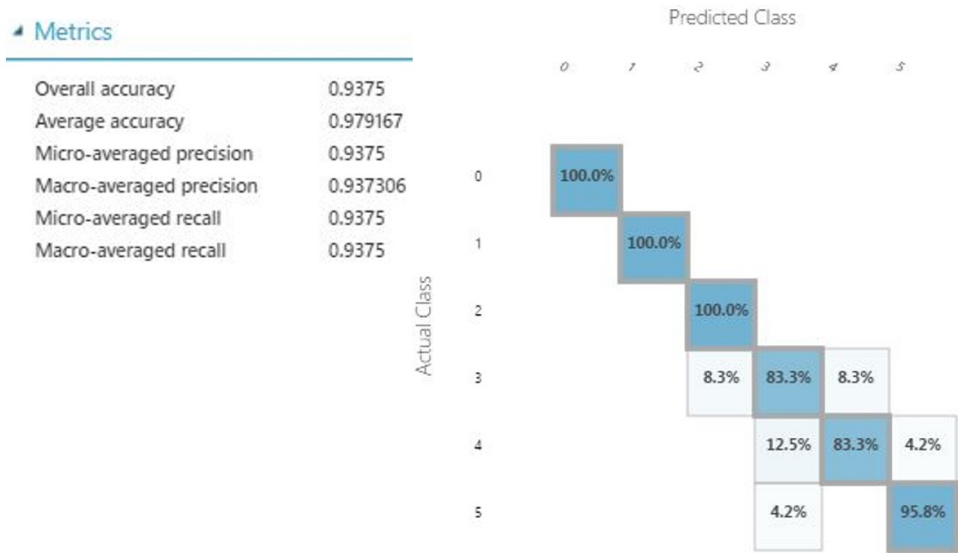

Figure 3.6: Classification confusion matrix of best parameterized CNN model in MAMLS for CT Image classification. Classification of image quality 3,4 ahve the highest errors.

for this data set is almost absolute ($\sim 99\%$). However, analysis of the AFMs obtained from the (C-S) layers at the end of model training is inspired from existing works in [7], [13], and [9]. The AFMs aid in identification of new features that may lead to improvements in overall classification accuracies.

The first feature that we learn from the AFMs is that spatial orientation is an important factor for image-based classification using CNN models. In Fig. 3.8, we observe the AFMs for the Face data set. Qualitative assessment of these AFMs presents two key observations. First, the region around eyes and nose are the regions of interest for classification purposes. This observation is intuitive with respect to existing facial recognition algorithms [31]. Second, the spatial orientation is a significant feature that impacts facial shape detection. Thus, each face is flipped by 180^o to double the number of data samples and perform classification for the MNIST and Face data sets.

For the CT image data set, 180^o flip does not alter the image variability significantly since the abdominal scans are spatially similar in the left and right regions. Hence, for this

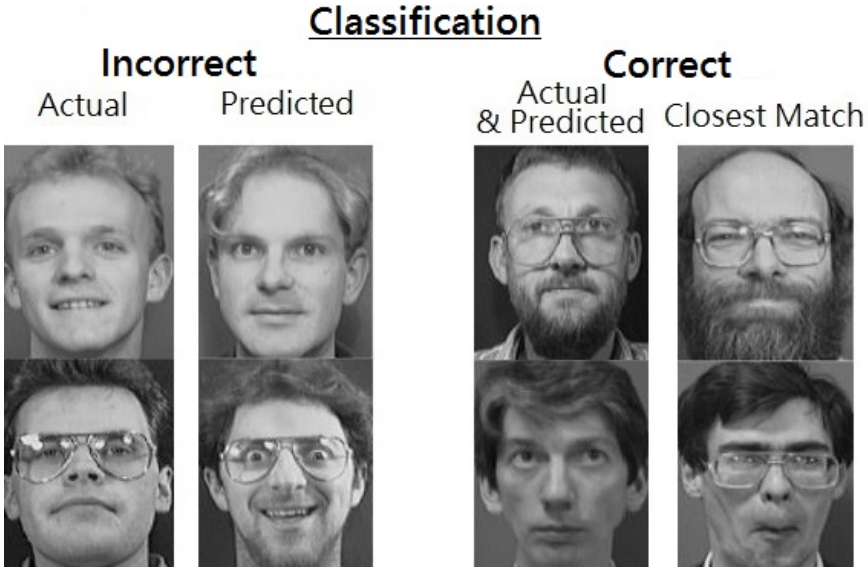

Figure 3.7: (Left column) represents miss-classified faces due to similar structural orientations. (Right column) represents borderline correct classifications owing to the structural similarity in facial occlusions and fields of view.

data set, each image is flipped by 90^o to increase image variability. It is noteworthy that the 70/30 data sample split is applied to the doubled sample size data sets such that no image from the test data set has a flipped version in the training data set, or vice versa. The impact of the doubled sample size by flipping the input images on the three data sets is presented in Table 3.5.

Table 3.5: Impact of image flipping on classification accuracy.

	MNIST small	FACE	CT Image
	$ACC_\mu(\%)$		
Original Samples	**99.2**	92.5	**93.7**
Doubled samples by flipping	98.75	**95.1**	84.7

From Table 3.5, we observe that image flipping enhances classification accuracy in the

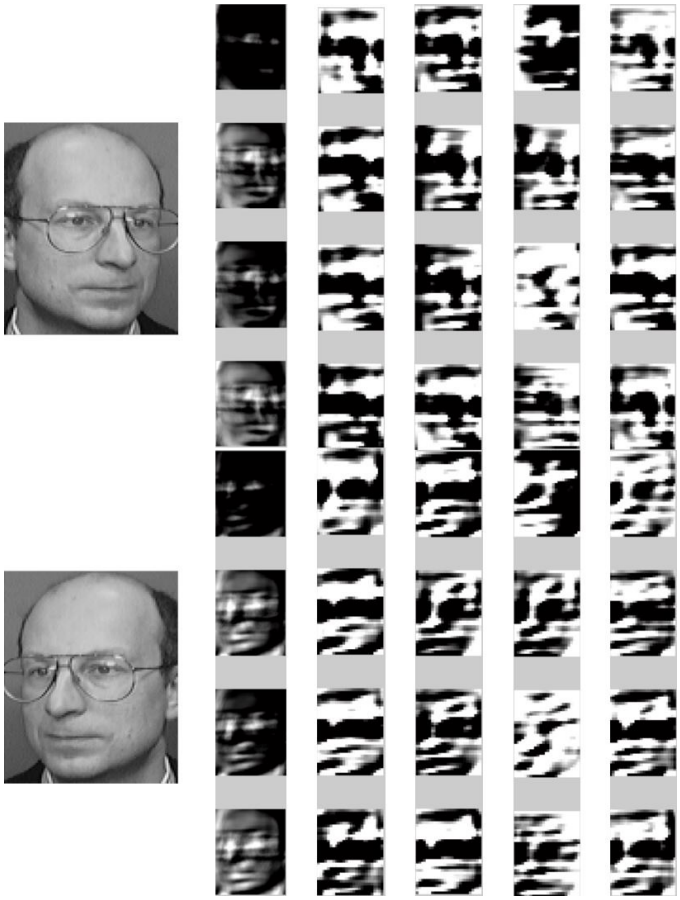

Figure 3.8: AFMs corresponding to original image (top) and 180^o flipped image (bottom). Image flipping increases sample size and variability in images, thus enhancing the overall ACC_μ.

Face data set only. This is because the CNN models are efficient in image shape detections. Thus, by flipping each facial image, the training image variability is increased. Hence, the CNN model is able to detect larger variations in image shape. For the MNIST data set, image flipping produces ambiguous samples, such as a flipped digit '5' is most likely to appear as a '2' as shown in Fig. 3.5. This ambiguity leads to lower ACC_μ for this data set with flipped image samples. For the CT image data set, most miss-classifications occur for image qualities 3, 4 as shown in Fig. 3.6. However, image flipping by 90^o distorts the organ shapes in the CT images, thereby resulting in lowered ACC_μ when compared to the original

set of samples. From this experiment, we infer that image flipping is not always the best resort for increasing the sample size. Instead, the activated regions from the AFMs should be analyzed to decide if image flipping is useful for a particular data set or not.

The other feature learned from the AFMs of the CT image data set include edge thickness. In Fig. 3.9 the first 12 AFMs corresponding to a CT image is presented. Here we observe that deeper layers extract mostly edge information for the CT image quality classification task. To rank order the contributions of AFMs, each AFM is resized to [0,1] pixel intensities and thresholded at a constant pixel value. The sum of all the pixels in the thresholded image is then correlated to the CT image quality (Y). The thresholded image from 5th AFM is observed to highly correlate with the CT image quality metric (correlation coefficient=0.8). This thresholded 5th AFM images for different image qualities is shown in Fig. 3.10.

Figure 3.9: AFMs for the CT image data sets demonstrate the importance of edge information on this data set.

Based on this observation, the edge information from each image is added to the original image to double the number of samples for CNN training and testing. In Table 3.6, the impact of doubling the image samples by adding the edges to the original images, as shown

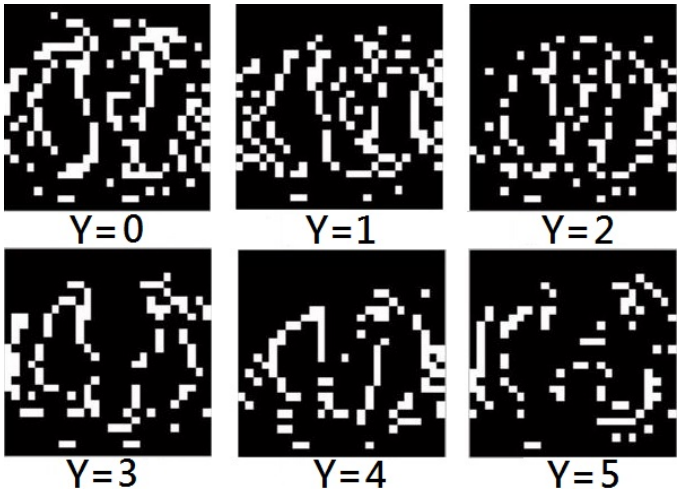

Figure 3.10: Thresholded 5th AFM images corresponding to the same CT slice from different image quality stacks. The thresholding pixel value=0.8. We observe that as the image quality improves (Y increases), the number of pixels in the thresholded image decreases. This observation is intuitive since low image quality induces large number of noisy false edges in the image.

in Fig. 3.5, is presented. Here, we observe that addition of edge information improves classification accuracy for the Face data set only. The reduction in classification accuracy with edge enhanced samples on the CT image data set is intuitive owing to the sensitivity of CNN models to structural patterns [5]. Since edge enhanced CT images extenuate the structural patterns in the image, the texture detection capability of the CNN model significantly reduces. Since CT image quality Y_k is dependent more on image texture than structure [5], the classification accuracy for edge enhanced CT images is lower than that of the original image set.

3.3.4 Importance Analysis of Model Parameterization and Training Data

The final experiment involves analyzing the importance of model parameterization on a highly variable data set. Based on prior works in [15], the number of training samples must be approximately 10 times the number of training parameters in the CNN model. For this

Table 3.6: Impact of image edge addition to the original image.

	MNIST small	Face	CT Image
	$ACC_{\mu}(\%)$		
Original Samples	**99.2**	92.5	**93.7**
Doubled samples by edge addition	98.75	**94.2**	90.8

experiment, we increase the number of data samples by combining all images from the three data sets by resizing each image to [119x119] pixels. The complete data set now contains a total of 60,886 sample images for data modeling using 70/30, 80/20, 90/10 training/test data splits, respectively. Next, we assign combined class labels to the MNIST small data samples with class labels [0:9], Face data set with class labels [10:49] and CT image class labels [50:55]. Finally, we re-categorize MNIST small samples with class label [0:9] as category 0, Face samples with class label [10:49] as category 1 and CT image class label [50:55] as category 2. The classification accuracies per category and per class within each category using kernels of size $[N\text{x}U\text{x}V]$ = [5x9x9], min-max normalization function and CNN architecture $[CS^2]$ are shown in Table 3.7.

Table 3.7: Performance Assessment of Combined Image data set on a fixed CNN model.

	MNIST small			Face			CT image		
Class Label	(0-9)			(10-49)			(50-55)		
Re-categorization	0			1			2		
Training/Test Split	70/30	80/20	90/10	70/30	80/20	90/10	70/30	80/20	90/10
ACC_{μ} Per Category (%)	100	100	100	100	100	100	100	100	100
ACC_{μ} Per Class-category (%)	86	83	81	25	3	9	33	33	17

From Table 3.7, we observe that the classification accuracy per category is consistently 100%. Thus, MNIST digits, Face images and CT images are always classified as themselves. However, the classification accuracy per class within each category is significantly less than the best classification accuracies obtained after training the model parameters and CNN architecture for each data set in Table 3.4. Also, classification per category is sensitive to training image data of the same nature. Thus, inter mixing image data sources to increase training samples does not improve overall classification accuracies. Specialized data sets must be treated separately for classification tasks and accurate data modeling.

Chapter 4

CNN MODEL FOR CT IMAGE QUALITY CLASSIFICATION

4.1 Introduction

Computed Tomography (CT) is a medical imaging modality that is mostly stored and accessed using the picture archiving and communication system (PACS). CT imaging is a vital diagnostic tool in modern health care with more than 70 million scans performed annually in 2007-2009 [32]. However, the rapid growth in the numbers of CT image scans performed over the past decade has raised concerns regarding patient health on account of exposure to radiation dosages [33]. Studies have shown that the extent of exposure to radiation for patients has doubled from 1996 to 2010 [34]. These observations necessitate CT radiation dosages to be maintained As Low As Reasonably Achievable (ALARA) while ensuring diagnostic image quality [35]. Some approaches to implement ALARA principles for CT imaging include minimizing the number of CT image acquisitions, optimizing CT image acquisition protocols, and improving low-dose CT image processing algorithms. Presently, the application of ALARA relies on simple image quality evaluation of non-anthropomorphic, unrealistic IQ phantoms or subjective evaluation of patient scans. We hypothesize that this evaluation could be improved with with Big Data analytics implemented using hardware independent frameworks developed on cloud-computing platforms [36].

Cloud-computing platforms provide enhanced storage and sharing capabilities for medical image data, allowing patient data to be accessed from multiple devices and different facilities [36]. This avoids the need for re-imaging and promotes collaboration with a wider range of medical experts [36]. Additionally, knowledge of past CT image acquisition parameters can help imaging specialists optimize patient-specific protocols, thereby resulting in dosage consistency for each patient. The cloud-computing frameworks enable distributed processing

of the large volumes of acquired CT images to implement existing, computationally intensive image processing algorithms and to develop and evaluate new image processing techniques [36][37]. For instance, one of the challenges posed by low-dose CT images is the low signal-to-noise ratio and several algorithms have been developed till date [38][39] to enhance low-dose CT image quality using small local datasets. However, with access to large sets of CT images, the cloud-computing platforms can be used to benchmark these methods for comparative analysis and standardization. This work is aimed at developing work-flows on cloud-computing platforms that can be useful for guidance of optimal CT image acquisition and processing protocols in accordance with ALARA while ensuring diagnostic CT image quality.

CNNs have been identified as a valuable Big Data analytics tool when raw image-based data is largely un-categorized [40]. In this work, we implemented CNN models in the MAMLS platform to classify CT images based on subjective CTIQ scores, thus bypassing the need for ROI selection and image segmentation. The CNN model learns the most significant hybrid features embedded in CT images from groups of pixels that are identified as feature maps by the convolutional and sub-sampling layers. The final neural network layer provides a probability score for the CTIQ based on the feature map regions. This cloud-based CNN work-flow overcomes CT image storage and processing limitations and leads to identification of structural patterns in CT images that can significantly benefit CT image acquisition parameter learning for each patient in personalized medicine applications.

4.2 Materials and Methods

4.2.1 CT Image Data

The data used for this work is obtained as abdominal scans from two distinct sources: phantoms and human patients. Each CT image comprises of [512x512] pixels. For the first set of phantom data, abdominal phantom CT scans are acquired by varying the image acquisition tube current values as [10; 25; 75; 125; 175; 350] mA, respectively, using a third

generation dual-source multi-detector CT scanner (Siemens SOMATOM FORCE; Siemens Healthcare AG, Erlangen, Germany). During image acquisition, most system settings are kept constant (helical, pitch 1.375, 20 mm collimation, 120 kV p). Corresponding to each tube current setting, 6 sets of 81 abdominal CT images are thus acquired. Since tube current values are directly proportional to the CTIQ, the subjective CTIQ varies as Y = [1; 2; 3; 4; 5; 6] for increasing tube current value, respectively. Thus, a multi-class classification task can be designed for the phantom data set. For the second set of patient data, from an IRB approved study, CT scans are acquired from 6 individual patients who were imaged in 2 separate sessions with different image acquisition parameters. The subjective CTIQ for each of the 12 sets of CT scans were assigned in range Y = [1 : 5], where Y = 1 implies poor quality image, Y = 3 implies optimal quality, and Y = 5 implies very high quality images, respectively. In this work, 100 images with [512x512] pixels acquired from the patient abdominal regions are analyzed from the 12 sets of patient image scans. It is noteworthy that the patient data images are limited to Y = [2, 3, 4] subjective CTIQ scores as opposed to the larger variations in CTIQ scores for the phantom data. This represents the lack of very poor and very high quality patient CT images, respectively. Thus, the following 3 classification tasks can be designed on the patient data set.

1. Multi-class Classification: This multi-class classification task, involves classification of each test image for Y = [2, 3, or 4], separately.

2. Low by High Binary Classification: This binary classification task, low by high (L/H), aims to distinguish CT images with low quality (Y = 2) from those with high quality (Y = [3, 4]).

3. Optimal by Sub-optimal Binary Classification: This binary classification task, optimal by sub-optimal (O/S), aligns with the goal of limiting radiation dosages to comply with ALARA while maintaining diagnostic image quality. In this task based on domain knowledge, CT images with Y$>$ 3 can be considered to have unnecessarily high quality

and are therefore sub-optimal. Thus, in this binary classification task CT images that are of optimal quality (Y = 3) are classified from those with sub-optimal CTIQ (Y = [2, 4]). For each of the three classification tasks, the mean classification accuracy is for classification performance evaluation.

4.2.2 Data Models

1. Baseline Model: This model [5] automatically locates a uniform ROI in each CT scan based on anatomical knowledge. An example of this model for an abdominal phantom image is shown in Fig. 4.1.

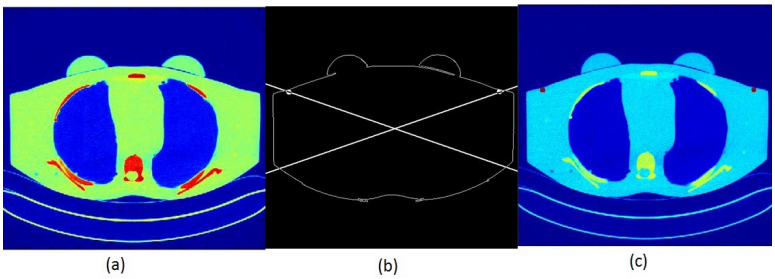

Figure 4.1: Steps for fixed-size ROI detection. (a) Phantom CT image. (b) Estimation of straight lines (horizontal angle θ) intersecting at the centroid of the abdominal region. (c) Patch(θ) regions detected as two circles with uniform area.

2. CNN Model: The CNN model is motivated by the prior work in [7], where each input image is subjected to several hidden layers of feature learning to generate an output vector of probability scores, for each image to belong to a certain CTIQ. For this analysis, we implement CNN architecture with the following 7 hidden layers: convolutional (C)-subsampling (S)-activation (A)-convolutional (C)-subsampling (S)-activation (A)-neural network (NN). Each C-layer convolves the input image with a set of kernels/filters, the S-layer performs pixel pooling, A-layer performs pixel scaling in the range [-1,1] and the final NN-layer implements classification using 200 hidden neurons. The CNN model has high computational complexity since it involves inten-

sive parameter training for the hidden layer feature maps and NN classifier weights and biases. To avoid computational bottlenecks, the CNN model is trained in the MAMLS platform followed by classification of the test sample images for CTIQ. For each test image, the CTIQ corresponds to the output class with maximum probability score assigned by the NN-layer. Prior works have shown that the finally trained feature maps from the hidden layers capture structural patterns embedded in images that aid image-based classification tasks [15]. In this work, we analyze these feature maps to gain new insights into the spatial configurations in CT images that aid CTIQ classification tasks. The CNN model is completely different from the other two models from the implementation standpoint, thereby allowing for potential identification of a more robust CTIQ estimation and classification method.

4.3 Experiments and Results

For analysis of the phantom and patient data sets, the following 3 sets of experiments are performed:

1. Multi-class Classification of Phantom Data Set: CTIQ classification for phantom data set is a multi-class classification task, where the data set is split such that 70% of all images are used for training and the remaining 30% images are used for model testing. The sample class frequencies for the training and test data sets are same. We observe that the mean accuracies for multi-class, CTIQ classification on the phantom data set are 0.736 and 0.938 using the baseline and CNN data models, respectively.

2. Cross-sectional Analysis and Classification of Patient Data Set: In this experiment using the patient data set, 1200 patient images are split with 70% and 30% of the images in the training and test data sets, respectively. These two sets are then used to train and test the 3 data models for multi-class, L/H, and O/S classification tasks. This cross-sectional analysis allows for comparison of the scalability for each model from the phantom to the patient dataset. Table 4.1 demonstrate the performance of

the CNN data model with respect to baseline model. Here, we observe that the CNN data model detects structural similarity in images for classification, thereby resulting in data over-fitting. Fig. 4.2 and Fig. 4.3 demonstrate the AFMs for high and lower quality CT images for a patient with high density fat and low density fat, respectively.

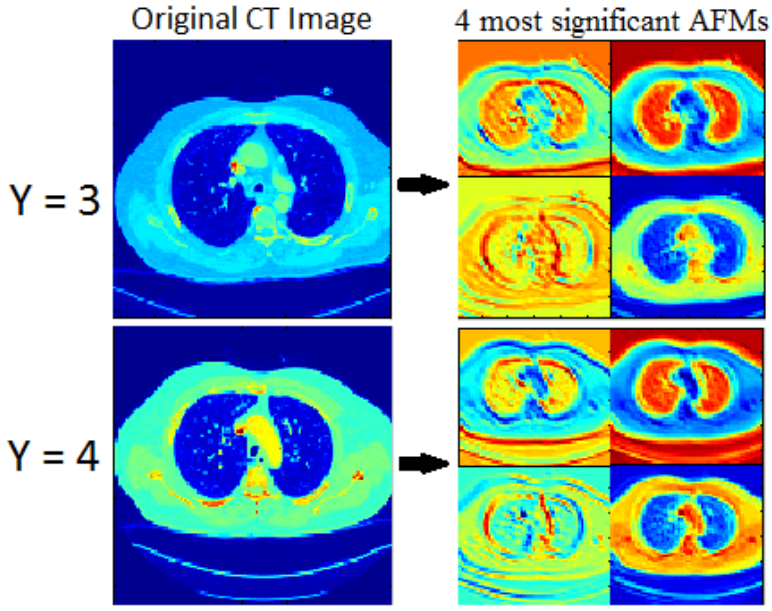

Figure 4.2: ROIs learned from trained feature maps in the CNN model.

Table 4.1: Mean accuracy of patient CT image classification for cross-sectional analysis using CNN.

Classification Task	Baseline	CNN
Multi-class	0.495	1
L/H	0.602	1
O/S	0.593	1

3. Longitudinal Analysis and Classification of Patient Data Set: In this experiment using the patient data set, one set of CT images from each of the 6 patients is used as the training set while the other set of CT images is the test set. This corresponds to a

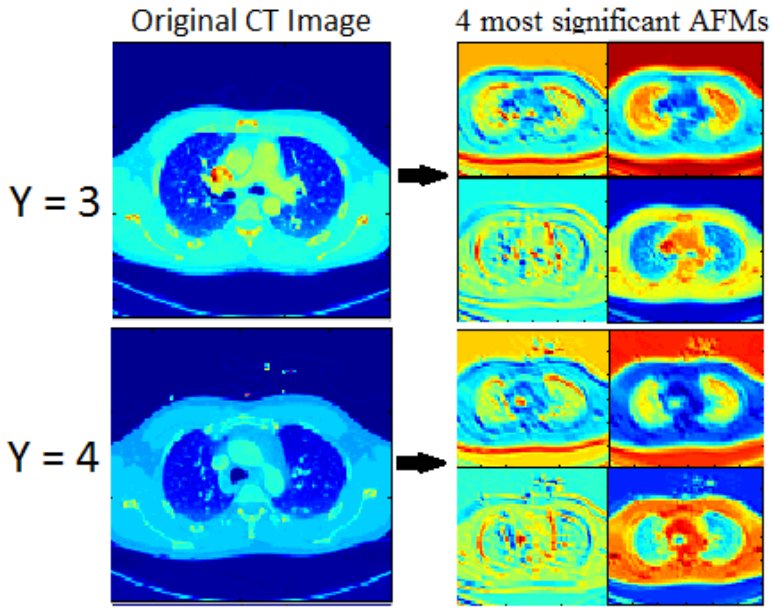

Figure 4.3: ROIs learned from trained feature maps in the CNN model.

50/50 data split. Using this training and test data sets, the 3 data models are analyzed for multi-class, L/H, and O/S classification tasks. This longitudinal analysis allows for assessment of CTIQ predictability for personalized medicine applications. Table 4.2 demonstrates the comparative performances of the CNN an baseline data models. Here, we observe that the baseline model has consistent classification performance in both the cross-sectional and longitudinal analyses.

Table 4.2: Mean accuracy of patient CT image classification for cross-sectional analysis using CNN.

Classification Task	Baseline	CNN
Multi-class	0.451	0.019
L/H	0.490	0.525
O/S	0.530	0.430

We observe that the phantom dataset varies only in CTIQ texture, but has consistent

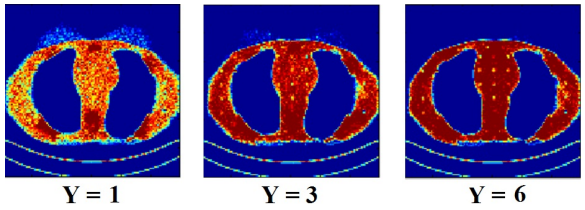

Figure 4.4: Phantom CT images with Y = 1, Y = 3, and Y = 6. All 3 images correspond to the same abdominal region in the phantom. We observe that CTIQ variations cause textural changes in phantom CT images.

structural information as shown in Fig. 4.4. Conversely, patient CT images have a higher degree of structural variability across patients and scanned instances as shown in Fig. 4.5 and Fig. 4.6, respectively. These additional variabilities in patient data result in the deterioration of classification performances for patient data in cross-sectional and longitudinal experiments. However, the structural dissimilarities between patient and phantom CT images lead to the variations in classification performances obtained from the CNN model. CNN models are known to be more sensitive to detection of structural similarities in images when compared to variations in image texture and granularity [15]. Thus, the CNN identifies structural similarities in the CT anatomy per patient rather than detect CTIQ. Hence, the CNN model is not preferable for CTIQ classification in the presence of vast image structural variabilities.

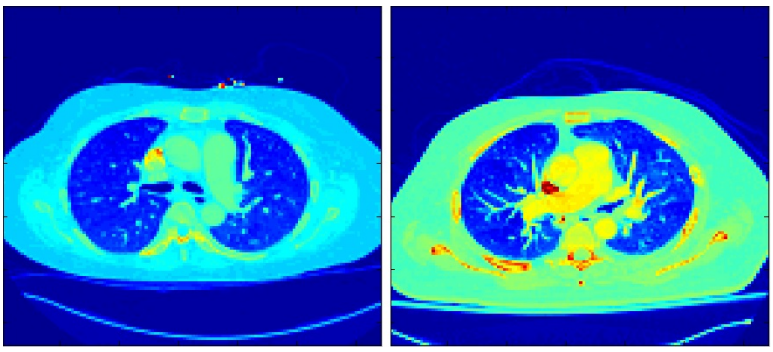

Figure 4.5: Structural variabilities observed across abdominal CT images across patients.

Additionally, the patient CT images are classified based on a subjective score (Y) that

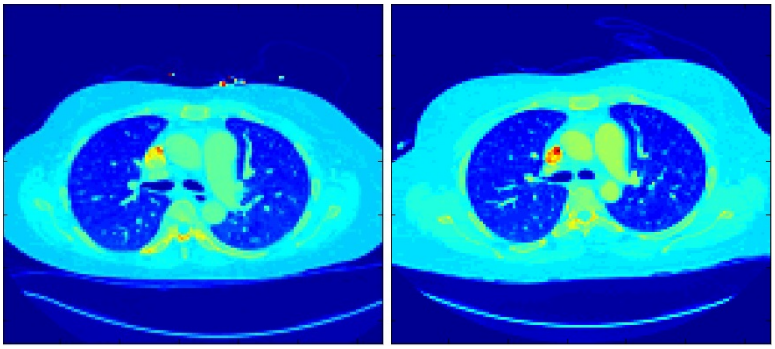

Figure 4.6: CT images of the same patient acquired at two different times, showing a change in alignment and sizes between the two imaging sessions.

is provided for a single image in each patient CT image scan set. While the rest of the CT image set is assumed to be the same quality, this assumption may not always hold. The image acquisition parameters vary throughout a patient scan to account for anatomical changes in the regions being scanned. These changes may result in different amounts of noise in two images acquired from the same scan. Thus, lack of subjective data results in over-fitted CNN models.

The CNN data model performs consistently on the phantom CT images and but it suffers from inconsistencies for the patient CT images. However, with the phantom CT images, the CNN proves a powerful tool in classifying CT image noise so long as the structural variabilities are low. Thus, the CNN-based classification performance may improve by masking the structural dissimilarities and by increasing the number of input images.

Chapter 5

CONCLUSIONS AND FUTURE WORKS

5.1 Conclusions

In conclusion, we present a novel CNN model that is parameterized for image-based data sets with different degrees of variations using grid search method. The parameterized CNN model by the proposed method is bench-marked with respect to existing state-of-the-art methods. It is observed that the proposed model performs similar to improved classification accuracies (up to 18% improvement in accuracy) on all the three multi-modal image data sets. Next, the importance of parameterization is analyzed with respect to varying kernel sizes and normalization functions. Here, we observe that the CNN predictive model is most sensitive to kernel size, normalization function and the CNN layering architecture. We propose a novel two-step process for best parameterization the CNN model and architecture that is generalizable across multi-modal image data sets.

One key contribution of this work is feature learning from the AFMs. Here, we analyzed two distinct features, image flipping and image edge enhancement. We observe that although image flipping and edge enhancement operations increase the numbers of data samples, these operations do not always enhance classification accuracies. For certain data sets such as the Face data set, the activated regions from the AFMs indicate that image orientation and image edges are significant decisive factors for facial recognition. However, image flipping and edge enhancement operations add ambiguity in the MNIST and CT image data sets, thereby reducing the overall classification accuracies. Thus, robust feature learning from AFMs is dependent on the nature of the input images.

Also, the importance of CNN model parameterization was analyzed by combining all the input images and using a fixed CNN parameter set. Here, we observe that deeper CNN

models are not always effective for finding new embedded features from images. For instance, in data sets with low to mild variabilities such as the CT image and Face data sets, deep CNN architectures (with > 11 layers) result in under-parameterized and over-fitted models with low classification accuracies, respectively. Thus non-deep CNN models parameterized using the proposed method are efficient in multi-class image-based classification tasks.

The impact of deep CNN models on large number of input image databases have been analyzed in [41]. However, the proposed non-deep CNN architectures are found to be favorable in the absence of large input data and varying degrees of structural and textural image variabilities.

In CTIQ classification task, CNN model implemented on a cloud-based platform of Microsoft Azure. Although the CNN model is the most computationally intensive, it best detects subtle variations in image granularity, that impact CTIQ, for phantom images. For the patient CT image multi-class classification, the baseline model has a mean accuracy 49.5% for the cross-sectional analysis and 45.1% for the longitudinal analysis. The CNN model over-fits the data, resulting in an accuracy of 100% for the cross-sectional analysis and 1.9% for the longitudinal study. The CNN models implemented on the MAMLS platform required 22-52 minutes of cloud computing time for phantom and patient image data sets for model training and testing operations. Further, analysis of the CNN model feature maps demonstrates detection of significant edges as features for CTIQ classification.

5.2 Future Work

The proposed method provides a novel work-flow for parameterizing CNN models across images acquired from multiple modalities. Future work will be directed towards varying combinations of kernel sizes, optimizing number of kernels, and analyzing their sensitivity to image classification tasks. Also, efforts will be directed to simulate occlusions on images to learn further embedded features from images that may lead to improved correlations for classification tasks.

AFMs have been found to detect structurally discriminating features across datasets.

However, all AFMs do not contribute equally towards classification tasks. Thus, there is a need to learn the importances of AFMs and rank them in order of classification significances. Such analyses will lead to lower the computational time complexities in CNN data models.

Next, the classification performance in medical image data can be improved by training with more labeled medical image samples. Future works can be directed towards improving the generalizabilities of the CNN data models. Better trained AFMs will lead to improve inferencing and protocol guidance. Also, This work in chapter 4 lends evidence to the hypothesis that CTIQ can be quantitatively measured and used for predictive modeling to a certain degree using rigorous and repeatable analytical strategies. Such quantitative analytics are necessary for guiding the present day diagnostic and care-delivery system.

Another future direction of work on CTIQ classification can be directed towards masking the structural variabilities across patient CT images and including image data extrapolation techniques to aid AFMs training and classification tasks. Additionally, 3D volumetric analysis of CT image stacks using 3D CNN models may enhance data inferencing.

So far, the CNN data model under investigation evaluates the global minimum for error minimization in the back-propagation function. Gradient descent has been utilized in the mathematical model used in this thesis. However, other optimization algorithms that evaluate multiple objectives for minimization/maximization applications can also be incorporated into the CNN data models. The use of genetic algorithm, evolution algorithm, and fuzzy optimization may lead to multi-objective local minimal that can offer new insights to image-based datasets.

BIBLIOGRAPHY

[1] Y. LeCun, C. Cortes, and C. Burges, "The mnist database of handwritten digits," 1998. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[2] B. C. Becker and E. G. Ortiz, "Evaluation of face recognition techniques for application to facebook," in *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on.* IEEE, 2008, pp. 1–6.

[3] K. Team, "Diabetic retinopathy winners interview: 4th place, julian and daniel," 2015. [Online]. Available: http://blog.kaggle.com/2015/08/14/diabetic-retinopathy-winners-interview-4th-place-julian-daniel/

[4] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on.* IEEE, 2015, pp. 922–928.

[5] S. Roychowdhury, N. Hollraft, and A. Alessio, "Blind analysis of ct image noise using residual denoised images," *arXiv preprint arXiv:1605.07650*, 2016.

[6] M. A. Nielsen, "Neural networks and deep learning," *URL: http://neuralnetworksanddeeplearning. com/.(visited: 01.11. 2014)*, 2015.

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[8] S. Hijazi, R. Kumar, and C. Rowen, "Using convolutional neural networks for image recognition," Tech. Rep., 2015. [Online]. Available: http://ip.cadence.com/uploads/901/cnn-wp-pdf

[9] M. Sirotenko, "Cnn - convolutional neural network class (https://www.mathworks.com/matlabcentral/fileexchange/24291-cnn-convolutional-neural-network-class)," in *MATLAB Central File Exchange*, 28 May 2009 (Updated 24 Oct 2012).

[10] M. Bihis and S. Roychowdhury, "A generalized flow for multi-class and binary classification tasks: An azure ml approach," in *IEEE International Conference on Big Data (Big Data).* IEEE, 2015, pp. 1728–1737.

[11] P. Ji, N. Zhao, S. Hao, and J. Jiang, "Automatic image annotation by semi-supervised manifold kernel density estimation," *Information Sciences*, vol. 281, pp. 648–660, 2014.

[12] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[13] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*. Springer, 2014, pp. 818–833.

[14] A. G. Howard, "Some improvements on deep convolutional neural network based image classification," *arXiv preprint arXiv:1312.5402*, 2013.

[15] D. L. Yamins and J. J. DiCarlo, "Using goal-driven deep learning models to understand sensory cortex," *Nature neuroscience*, vol. 19, no. 3, pp. 356–365, 2016.

[16] S. Gunasundari and S. Baskar, "Application of artificial neural network in identification of lung diseases," in *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*. IEEE, 2009, pp. 1441–1444.

[17] Y. Chen, L. Shi, Q. Feng, J. Yang, H. Shu, L. Luo, J.-L. Coatrieux, and W. Chen, "Artifact suppressed dictionary learning for low-dose ct image processing," *IEEE transactions on medical imaging*, vol. 33, no. 12, pp. 2271–2292, 2014.

[18] V. Hegde and R. Zadeh, "Fusionnet: 3d object classification using multiple data representations," *arXiv preprint arXiv:1607.05695*, 2016.

[19] N. S. Alvar, M. Zolfaghari, and T. Brox, "Orientation-boosted voxel nets for 3d object recognition," *CoRR*, vol. abs/1604.03351, 2016. [Online]. Available: http://arxiv.org/abs/1604.03351

[20] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1912–1920.

[21] R. Socher, B. Huval, B. Bath, C. D. Manning, and A. Y. Ng, "Convolutional-recursive deep learning for 3d object classification," in *Advances in Neural Information Processing Systems*, 2012, pp. 665–673.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[23] D. Y. Li Deng, "Deep learning: Methods and applications," Tech. Rep., May 2014. [Online]. Available: https://www.microsoft.com/en-us/research/publication/deep-learning-methods-and-applications/

[24] D. Mo, "A survey on deep learning: one small step toward ai," *Dept. Computer Science, Univ. of New Mexico, USA*, 2012.

[25] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE transactions on neural networks*, vol. 8, no. 1, pp. 98–113, 1997.

[26] V. Cherkassky and F. Mullier, "Learning from data," *John Wiley and sons, New York*, 1998.

[27] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on.* IEEE, 1994, pp. 138–142.

[28] D. Maji, A. Santara, P. Mitra, and D. Sheet, "Ensemble of deep convolutional neural networks for learning to detect retinal vessels in fundus images," *arXiv preprint arXiv:1603.04833*, 2016.

[29] M. D. McDonnell, M. D. Tissera, T. Vladusich, A. van Schaik, and J. Tapson, "Fast, simple and accurate handwritten digit classification by training shallow neural network classifiers with the extreme learning machinealgorithm," vol. 10, no. 8. Public Library of Science, 2015, p. e0134254.

[30] G. Loosli, S. Canu, and L. Bottou, "Training invariant support vector machines using selective sampling," *Large scale kernel machines*, pp. 301–320, 2007.

[31] S. Roychowdhury and M. Emmons, "A survey of the trends in facial and expression recognition databases and methods," *arXiv preprint arXiv:1511.02407*, 2015.

[32] D. B. Larson, L. W. Johnson, B. M. Schnell, S. R. Salisbury, and H. P. Forman, "National trends in ct use in the emergency department: 1995–2007 1," *Radiology*, vol. 258, no. 1, pp. 164–173, 2011.

[33] R. Smith-Bindman, D. L. Miglioretti, and E. B. Larson, "Rising use of diagnostic medical imaging in a large integrated health system," *Health Affairs*, vol. 27, no. 6, pp. 1491–1502, 2008.

[34] R. Smith-Bindman, D. L. Miglioretti, E. Johnson, C. Lee, H. S. Feigelson, M. Flynn, R. T. Greenlee, R. L. Kruger, M. C. Hornbrook, D. Roblin *et al.*, "Use of diagnostic imaging studies and associated radiation exposure for patients enrolled in large integrated health care systems, 1996-2010," *Jama*, vol. 307, no. 22, pp. 2400–2409, 2012.

[35] R. Smith-Bindman, "Is computed tomography safe," *N Engl j Med*, vol. 363, no. 1, pp. 1–4, 2010.

[36] G. C. Kagadis, C. Kloukinas, K. Moore, J. Philbin, P. Papadimitroulas, C. Alexakos, P. G. Nagy, D. Visvikis, and W. R. Hendee, "Cloud computing in medical imaging," *Medical physics*, vol. 40, no. 7, p. 070901, 2013.

[37] S. Roychowdhury and M. Bihis, "Ag-mic: Azure-based generalized flow for medical image classification," *IEEE Access*.

[38] Y. Nakayama, K. Awai, Y. Funama, M. Hatemura, M. Imuta, T. Nakaura, D. Ryu, S. Morishita, S. Sultana, N. Sato *et al.*, "Abdominal ct with low tube voltage: preliminary observations about radiation dose, contrast enhancement, image quality, and noise 1," *Radiology*, vol. 237, no. 3, pp. 945–951, 2005.

[39] Z. S. Kelm, D. Blezek, B. Bartholmai, and B. J. Erickson, "Optimizing non-local means for denoising low dose ct," in *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. IEEE, 2009, pp. 662–665.

[40] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 2, no. 1, p. 1, 2015.

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[42] L. Kang, P. Ye, Y. Li, and D. Doermann, "Convolutional neural networks for no-reference image quality assessment," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1733–1740.

[43] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[44] D. Karmakar and C. Murthy, "Face recognition using face-autocropping and facial feature points extraction," in *Proceedings of the 2nd International Conference on Perception and Machine Intelligence*. ACM, 2015, pp. 116–122.

[45] "5 ways companies are using big data to help their customers," 2014. [Online]. Available: http://venturebeat.com/2014/04/21/5-ways-big-data-is-helping-companies-help-their-customers/

[46] S. Deng, Y. Tian, X. Hu, P. Wei, and M. Qin, "Application of new advanced {CNN} structure with adaptive thresholds to color edge detection," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 4, pp. 1637 – 1648, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1007570411004989