VizioMetrics: Mining the Scientific Visual Literature

Po-Shen Lee

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2017

Reading Committee:

Linda Shapiro, Chair

Bill Howe, Co-chair

Jevin D. West

Program Authorized to Offer Degree:
Electrical Engineering

University of Washington

Abstract

VizioMetrics: Mining the Scientific Visual Literature

Po-Shen Lee

Chair of the Supervisory Committee:
Professor Linda Shapiro
Department of Electrical Engineering

Scientific results are communicated visually in the literature through diagrams, visualizations, and photographs. In this thesis, we developed a figure processing pipeline to classify more than 8 million figures from PubMed Central into different figure types and study the resulting patterns of visual information as they relate to scholarly impact. We find a significant correlation between scientific impact and the use of visual information. Moreover, we find that citations within the same field tend to correlate with tables while citations from other fields tend to correlate with diagrams, suggesting that visual representations aid interdisciplinary communication. These results suggest that encoding results visually improves communicability, but these visual elements remain ensconced in the surrounding paper and difficult to use directly to facilitate information discovery tasks or longitudinal analytics. Very few applications in information retrieval, academic search, or bibliometrics make direct use of the figures, and none attempt to recognize and exploit the *type* of figure, which can be used to augment interactions with a large corpus of scholarly literature.

We use these results to articulate a new research agenda "viziometrics" to study the organization and presentation of visual information in the scientific literature. We present VizioMetrics.org, a platform that extracts visual information from the scientific literature and makes it available for use in new information retrieval applications and for studies that look at patterns of visual information across millions of papers. The VizioMetrics.org processes a corpus of documents, classifies the figures, organizes the results into a cloud-hosted database, and drives three distinct applications to

support bibliometric analysis and information retrieval. The first application supports information retrieval tasks by allowing rapid browsing of classified figures. The second application supports longitudinal analysis of visual patterns in the literature and facilitates data mining of these figures. The third application supports crowdsourced tagging of figures to improve classification, augment search, and facilitate new kinds of analyses. In addition, we proposed PhyloParser, an end-to-end framework for automatically extracting species relationships from phylogenetic trees using a multi-modal approach to digesting diverse tree styles. PhyloParser enables extraction of phylogenies from a large scale of dendrograms. As an extended application of VizioMetrics.org, we aim to build a public database of phylogenetic information that covers the historical literature as well as current data, and then use it to identify areas of disagreement and poor coverage in the biological literature.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I am deeply grateful to my thesis advisor Prof. Bill Howe. He always steers me in the right direction whenever I needed it and gives me the greatest support to my graduate life. I also want to express my gratitude to Prof. Jevin D. West for the continuous support of my Ph.D study and Prof. Linda Shapiro for guiding me in the world of computer vision. Besides my advisors, I would like to thank the rest of my thesis committee members Prof. Jenq-Neng Hwang and Prof. John D Wilkerson for their encouragement, insightful comments, and hard questions. I would particularly like to thank Prof. Richard Ladner, who opened the door for me in my most difficult days in the graduate school. I would not be able to complete my degree without your support.

I also thank Sean Yang and Maxim Grechkin for the contribution in our collaborated work and also bring VizioMetrics to a great height. To other labmates in DataLab and GRAIL: Yeaseul Kim, Jason Portenoy, Ian Wesley-Smith, Ezgi Mercan, Deepali Aneja, Yao Lu, Sachin Mehta, Shima Nofallah, Shu Liang, thank you all for your valuable advices on my research. In particular, I am grateful to the generous support from Lia Kazakova, Bum Mook Oh, who helped with my project as undergrad researchers.

I would also like to thank the authors and publishers who grant the permission of using their figures for presenting my work and all authors who ever involved in the survey of VizioMetrics.

Finally, I must express my very profound gratitude to my parents and to my wife Chloe for providing me with unfailing support and continuous encouragement throughout the hairpin turn of my study area and through the process of researching and writing this thesis.

This accomplishment would not have been possible without them.

Po-shen Lee

May-30th-2017

Chapter 1

INTRODUCTION

Information in the scientific literature is conveyed visually using plots, photographs, illustrations, diagrams, and tables. This information is designed for human consumption but, unlike the surrounding text, is not directly machine-readable. As a result, relatively few studies explore how these visual encodings are used to convey scientific information in different fields and how patterns of encodings relate to impact.

The visual cortex is the highest-bandwidth information channel into the human brain [98], and humans are known to better retain information presented visually [69]. The figures in the scientific literature, therefore, would appear to play a critical role in scientific communication. The discovery of the structure of DNA was largely a visual argument based on the images produced by X-ray crystallography; indeed, Gibbons argues that the act of producing the visualization of the structure represents the discovery itself [33]. The first extra-solar optical images of planets amplified the nascent subfield of astronomy focused on planet-hunting [49]. Medical imagery of biological processes at scales below that which can be detected using conventional optical methods are providing new insight into brain function [23]. In all fields, key experimental results are summarized in plots, complex scientific concepts are illustrated schematically in diagrams, and photographic evidence are used to provide insight at scales and in locations not available to the human eye. The quantification of science and the rise of big data has increased the need for visual representations of the data, models, and results.

In the 1950s, researchers like Eugene Garfield and De Solla Price recognized the importance of citations in organizing and searching the scientific literature [25, 32], but the process for making this information useful at scale was painstaking. We see an analogy with the current role of the visual literature. There is clear value in extracting and analyzing figures to understand its role in

scientific communication and impact, just as there is clear value in analyzing the citation network in isolation. The citation network tells us how ideas are related; visual representations tell us how ideas are communicated. Figures from related groups, authors, and fields share a 'DNA' that can reveal how information is conveyed.

We adopt the term *viziometrics* to describe this line of research to convey the shared goals with bibliometrics and scientometrics. As with bibliometrics, viziometrics uses citations to measure impact, but focuses on relating impact to the patterns of figure use. We analyze theses patterns within the papers (specifically, the distribution of various figure types) in order to understand how they may be used to more effectively communicate ideas. We have two overarching goals, towards which this study represents an initial step: First, we seek to build new tools and services based on the visual information in the literature to help researchers find results more efficiently. For example, when searching for uses of a particular method (e.g., phylogenetic analysis), the figures themselves are more relevant than the papers that contain them. Second, can the patterns of figure use inform new best practices for scientific communication, especially outside of the authors' own discipline?

In this thesis, we present an initial exploration of viziometrics by analyzing a corpus of papers from PubMed Central to relate the use and distribution of visual information. We found that the number of diagrams and plots per page correlates positively with article influence. A possible interpretation is that clarity is critical for impact: illustrating an original idea may be more influential than quantitative experimental results. In addition, we find that citations from within the same field tend to correlate with tables, while citations from other fields tend to correlate with diagrams, suggesting that visual representations aid interdisciplinary communication. Based on this finding, we described a new application to search and browse scientific figures, potentially enabling new kinds of search tasks. The VizioMetrics.org systems affords search by keyword as well as figure type, and shows results in a figure-centric layout. We also encourage people to use our publicly available corpus and software to explore this area of research and create a new community of interest.

Our key contributions are:

- We present an image processing pipeline that classifies scientific figures into different categories. We develop computer vision techniques for identifying multi-chart figures and dismantling them into singleton figures before classification.

- We build a search interface that uses the classified images as the primary unit for exploring scholarly content.

- We build a crowdsourced platform for labelling scientific figures.

- We present an automatic algorithm to extract phylogenetic data from dendrograms, toward a retrieval service for genetics and evolution research based on massive phylogenetic data extracted from millions of dendrograms.

- We analyze the relationship between the use of visual information and measures of impact, concluding that highly cited papers tend to include more diagrams, plots and tables than ordinary papers. In addition, we find that papers with certain figure types that explain abstract concepts or new models and methods are more cited from other fields than papers with just tables displaying data.

- We release a fully-annotated dataset for public in order to support additional analyses of the figures and improve figure-oriented search.

Chapter 2

RELATED WORK

Benefitting from a powerful search engine, such as Google Scholar and Microsoft Academic Search, as well as rich archives from digital libraries such as PubMed, PSU CiteSeerX and Cornell arXiv, researchers are able to collect papers more efficiently than in the past. In the traditional way of searching, users give titles, authors, or related keywords as inputs. A paper list ordered by matching score or impact factor can be obtained within a second. This method has been used over twenty years, and the search algorithms are improved in accuracy. However, graphic information in literature has not been widely considered as metadata or even search targets. On the other hand, content-based image retrieval (CBIR) organizes digital image archives by their visual content allowing users to retrieve images sharing similar visual elements with query images [24, 62, 87]. This technology has been widely deployed and is available in multiple online applications. However, CBIR has not been used to enhance scientific and technical document retrieval, despite the importance of figures to the information content of a scientific paper.

Scientific results are communicated visually in the scientific literature, but visual information is underused in academic search tools. Recently, by the foreseeable purposes for searching scientific figures as well as mutual techniques in computer vision, image retrieval, databases, etc., figure retrieval starts sprouting in this area. In this section, first we survey the related studies of computer vision for classifying and understanding figure images, second we report the early projects of figure retrieval, and third we introduce previous studies in bibliometrics and scientometrics that share the same idea of "viziometrics."

2.1 Computer Vision for Mining Visual Literature

Computer vision problems for mining visual literature can roughly be categorized to (1) classifying scientific figures and (2) extracting information from such figures.

2.1.1 Figure Classification

In early studies, designing appropriate feature descriptors for figure images is the main approach to solving the problem of classifying figure images. Futrelle et al presented a diagram-understanding system utilizing graphics constraint grammars to recognize two-dimensional graphs [30]. Later, they proposed a scheme to classify vector graphics in PDF documents via spatial analysis and graphemes [31,84]. N. Yokokura et al presented a layout-based approach to build a layout network containing possible chart primitives for recognition of bar charts [107]. Y. Zhou et al used Hough-based techniques [109] and Hidden Markov Models [110] to approach bar chart detection and recognition. W. Huang et al proposed model-based method to recognize several types of chart images [44]. Later they also introduced optical character recognition and question answering for chart classification [43]. In 2007, V. Prasad et al applied multiple computer vision techniques including Histogram of Orientation Gradient, Scale Invariant Feature Transform, detection of salient curves etc. as well as Support Vector Machine (SVM) to classified five commonly used charts [75]. In 2011, Savva et al. proposed an application of chart recognition [82]. Their system classifies charts first, extracts data from charts second and then re-designs visualizations to improve graphical perception finally. They achieved above 90% accuracy in chart classification for ten commonly used charts.

Since 2012, feature-based approaches have been gradually replaced by data-driven approaches for classification problems in the computer vision area. Convolutional Neural Networks (CNNs) has emerged as the state-of-the-art models for natural image classification [39,54,86]. Encouraged by the tremendous success of CNNs, people started to investigate the use of CNNs in classifying figure images. In recent study, Siegel et. al fine-tuned the pretrained AlexNet [54] and ResNet-50 [39] with 60,000 figures and classified them into seven categories. They report mean accuracies

of 84% and 86% respectively from AlexNet and ResNet-50. In our study, we also obtain a better result using convolutional neural networks compared to the state-of-the-art [82].

None of these efforts involved the use of multi-chart images. In this study, we introduce a complete figure processing pipeline for scientific figures. It includes a composite figure classifier to identify multi-part figures, a composite figure dismantler to segment multi-part figures, and a figure-type classifier to automatically label figures.

2.1.2 Information Extraction From Visual Literature

Line graphs are the most commonly used data-driven visualization. A research team from Pennsylvania State University has been engaged in mining line graphs for decades. Lu et al. proposed an automated algorithm for extracting components form 2-D line curves [65]. They report a match rate of 72.5% by manually compare 40 plots and the corresponding redrawn plots. Later in 2008, Kataria et al. from the same team extended the algorithm to handle scatter and curve-fitted plots and also proposed a text block detection algorithm to parse tick labels and legends [13, 51]. In 2015, the team integrated their techniques and proposed an automatic algorithm for extracting raw data from line graphs [18].

In 2016, another team from the Allen AI Institute addressed the same problem using a different approach powered by deep learning. Siegel et al. proposed a thorough pipeline for extracting data from curve plots [85]. Their pipeline integrates several modules such as axes parser, legend parser, and curve parser. For the data extraction part, they trained a Siamese network to extract similarity features of curves and then use rank SVM formulation with boosting to find best path of each curve. Their approach achieve an overall accuracy of only 17.3%, because several modules all need to be accurate for the entire parsing to be considered perfect.

Instead of parsing line graphs, Savva et al. proposed an algorithm to extract data from bar charts and pie charts [82]. Their method achieved impressive result but it highly depends on manually locating text and limits to plots with colors.

This related work focuses on plots that present quantitative data. Despite focusing on specific type of figures, it is still a very challenging problem for their diverse styles. Furthermore, parsing

diagrams that illustrate methodologies, mechanisms, and relationships is even more challenging and rarely studied. Compared to plots, information embedded in diagrams is more likely to be difficult to describe solely by text, for instance, *a metabolic pathway illustrating linked series of chemical reactions occurring within a cell* [1] and *a phylogenetic tree showing the inferred evolutionary relationships among various biological species* [2]. In 2016 Kembhavi et al. proposed Diagram Parse Graphs, a first model for parsing and studying semantic interpretation of diagrams that illustrates relationships between nature objects [53]. They collected 5,000 diagrams from science textbooks used in elementary schools and developed models for syntactic parsing and question answering.

The two related studies [53, 85] published in 2016 reveal the increasing attention of understanding visual illustration in the computer vision community. Compared to natural images, visual illustrations encode rich knowledge that has been well organized, which are potentially sources for artificial intelligence study.

In this study, we dive into phylogenetic trees, a specific type of diagram widely used to illustrate the result of an evolutionary analysis. In 2017, there are 206,764 articles returned in PubMed when searching "phylogen*". A. Rambaut proposed the first interactive tool to convert a tree image into machine-readable format [76]. It requires the user to reconstruct the tree by clicking on each of its nodes in turn. In 2007, Laubach et al. proposed TreeSnatcher, a semi-automatic application to recover the phylogenetic data under the user's supervising in GUI [57]. Later in 2012, they upgraded the application with more interactive tools of image processing and drawing function [58]. The semi-automatic application effectively reduces the consuming time for accurately parsing a tree. However, it is not a solution for parsing a large collection. In 2011, Hughes proposed TreeRipper, a web application that automatically converts the tree image into NEXUS, Newick and phyloXML formats [45]. The application aims at both bifurcating and multifurcating trees, but it has strict prerequisites for the style of input images. The algorithm starts with a sequence of heuristic cleaning steps followed by tracing the contour of the tree topology. The author reports

[1] defined by Wikipedia

[2] defined by Wikipedia

37 successful samples out of 114 phylogenetic tree diagrams. As the pioneer of parsing tree, TreeRipper shares the same goal with our study; however, it is not tolerant to noise, which is very crucial to the contour tracing and restricts the accuracy of TreeRipper.

2.2 Figure Retrieval

Computer vision techniques have been used in the context of conventional information retrieval tasks (retrieving papers based on keyword search), including some commercial systems such as D8taplex [2] and Zanran [1]. Search results from these proprietary systems have not been evaluated and do not appear to make significant use of the semantics of the images.

In 2001, Murphy et al. proposed a Structured Literature Image Finder (SLIF) system, targeting microscope images [67]. A decade later, Ahmed et al. [4, 5] improved the model for mining captioned figures. The latest version combines text-mining and image processing to extract structured information from biomedical literature. The algorithm first extracts images and their captions from papers, then classifies the images into six classes. Classification information and other metadata can be accessed via web service. However, SLIF focuses exclusively on microscropy images and does not extend to general figures.

Choudhury et al. [77] proposed a modular architecture to mine and analyze data-driven visualizations that included (1) an extractor to separate figures, captions, and mentions from PDF documents [19], (2) a search engine [8], (3) raw-data extractor for line charts [13, 18, 51, 65], and (4) a natural language processing module to understand the semantics of the figure. Also, they presented an integrated system from data extraction to search engine for user experience. Chen et al. [16] proposed their search engine named DiagramFlyer for data-driven figures. It recovers the semantics of text components in the statistical graph. Users can search figures by giving attributes of axes or the scale range in further. Additionally, DiagramFlyer can expand queries to include related figures by expanding the keywords using a lexicon generator. Other studies have proposed informatics methods for retrieving maps of the brain through large-scale image and text mining on fMRI images [74].

These early studies built their search engines on top of pre-selected images in particular uses.

Our approach is to classify figures based on their type, then organize a search and analysis interface that uses these classified images as the primary unit of interaction.

2.3 Visual Patterns of Science

Although the early projects represent a different approach for information retrieval tasks, they make no attempt to analyze the patterns of visual information in the literature longitudinally.

Decades ago, Lindsey reported a positive correlation (+0.33) between graph use and citation rate in psychology [63]. Cleveland surveyed 57 journals in various disciplines and showed a significant difference of graph use across disciplines and journals, identifying patterns associated with specific journals [20]. These studies identified the importance of the visual literature, but represent small-scale studies and do not reflect changing trends in the technology used to create visualizations and the role of interdisciplinary work.

In more recent studies, Hegarty et al. analyzed 1,133 articles published in 9 leading psychology journals, finding that articles with fewer graphs and more equation models were more frequently cited [40]. This result did not hold in other studies outside of psychology. Fawcett et al. found that the heavy use of equations has a significant negative impact on citation rates in three journals: *Evolution*, *Proceedings of the Royal Society of London B: Biological Sciences* and *The American Naturalist* [28]. Tartanus et al. analyzed all papers published in 2010 from 21 selected journals in agriculture [91] and report a positive correlation between number of graphs and two-year impact factors in journal level. Cabanac et al. studied 5,180 articles from six science and social science journals and found that groups of authors used significantly more tables and graphs than single authors [14].

Hartley et al. investigated approximately 2,000 articles from 200 journals in the sciences and social sciences. They found that men used 26% more figures than women, but found no significant difference in their use of tables. In addition, they didn't find significant differences between men and women in using either graphs and figures or tables in social science articles [38]. Since counting figures manually is extremely time-consuming, all of these studies were limited to specific domains on a relatively small number of papers and journals. Our approach is to automate

the analysis using computer vision techniques and machine learning, scale it to a large corpus of papers to allow broader inferences, and release the software and labeled data for other researchers to use.

These studies focus on different disciplines, but most provide evidence that communicating results visually is associated with increased citations. However, such studies were restricted to limited samples in selected domains, corroborating Cleveland's finding and our own work that the role of visual communication varies across domains. Most importantly, none of these previous studies consider how impact varies across figure types, which indicate the type of information, for instance, diagrams are usually used for illustrating methodologies and concepts and plots are usually used for presenting quantitative results. It allows us to investigate the correlation between article impact and presentation methods. We also show how these differences change when looking at citations across and within fields. It provides an evidence that different audiences respond to different presentation methods.

Chapter 3

VIZIOMETRICS DATA PREPROCESSING

Scholarly papers are often subject to a paywall that restricts access to users who have paid a fee. The unavailability and copyright increases the access barrier for building a search engine for academic papers and become even more difficult for the embedded figures. Limited by the barrier of the paywall, we compiled a scholarly dataset from free-access archives. We chose PubMed Central (PMC) as our main source because it includes a large collection of papers (1 million+) in life science and provides free access to the full text documents including the source images.

3.1 Acquiring Figure Images

We downloaded 1,022,319 article files from the PMC FTP server and extracted the images into a figure corpus. Of these files, about 66% had associated figure files. These figure files had been perviously separated from the PDF files, allowing us to avoid having to extract them from the literature. In total, we obtained 8,913,298 images from 681,630 papers. In addition, we extracted the paper metadata from the full text documents, including paper ID, doi, source journal, paper title, authors, publishing date, citations, pages and image captions that we use in our figure search engine and analysis.

We found five image formats in use: GIF, JPEG, TIF, TIFF, PNG. The vast majority (99%) of the images were in JPEG format with a small number of PNG files. We had several filtering steps to remove duplicate images and the images that are not scientific figures (e.g. copies of full papers). First, we removed all GIF files since they are duplicates of images in other formats. Second, we removed image files that turned out to be image representations of full papers using regular expression to match their particular string pattern. Third, we converted all TIF and TIFF files to JPEG files and resized their dimensions such that the longer edge was 1280 pixels. If the

longest edge of the original image was larger than this value, we did not modify the aspect ratios. After filtering, we obtained 4,781,742 images.

3.2 Classifying Figure Images

After filtering, we classified 4,781,742 images into five categories. The classification algorithm is described in Section 4. The classifier returns a probability distribution across all class types, but for each image we only assigned the label with the highest probability. The class labels are as follows:

- Equation (e.g., embedded equations, Greek and Latin characters)
- Diagram (e.g., schematics, conceptual diagrams, flow charts, architecture diagrams, illustrations)
- Photo (e.g., microscopy images, diagnostic images, radiology images, fluorescence imaging)
- Table (any tabular structures with text or numeric data in the cells)
- Plot (e.g., bar charts, scatter plots, line charts)

Of the 4.8 million figures, 1.4 million contained multiple sub-figures within a single image, often with each sub-figure labeled with A, B, etc. We refer to these figures as *multi-chart* figures. We "dismantled" these multi-chart figures into their individual parts using a customized algorithm that we developed for this purpose (Section 4.1). After dismantling, we extracted and classified another 5 million individual figures. In total, we classified more than 10 million figures.

The results of our classification are summarized in Table 3.1. This summary information alone provides some interesting insights: About 67% of the total figures are embedded in multi-chart figures, demonstrating the importance of dismantling figures for this analysis. Plots are the most likely figure type to be embedded in this way: we found 475k standalone plots but 3.5M total plots after dismantling. There is a relatively uniform distribution across diagrams, photos, and plots; the prevalence of photos is likely an artifact of the biomedical emphasis of the PMC corpus.

Table 3.1: We classified 4,781,741 figures into six categories. The table shows the number of figures for each figure type before and after dismantling.

Figure Type	Count Before Dismantling	Count After Dismantling
Multi-chart	1,416,237 (29.6%)	None
Equation	1,425,042 (29.8%)	1,741,059 (17.0%)
Diagram	652,918 (13.7%)	2,036,704 (19.9%)
Photo	475,615 (9.9%)	2,322,231 (22.7%)
Plot	475,327 (9.9%)	3,579,839 (35.0%)
Table	336,602 (7.1%)	553,171 (5.4%)
Total	4,781,741	10,233,004

The classification results together with figure metadata (captions, image url, image format, etc) are stored in our database for further use.

3.3 Predicting Key Figures

An article abstract exists in almost every scientific paper, helping readers communicate the key points of an article. However, text is not always the most efficient and effective way to describe an abstract concept or a complex mechanism. In 2010, *Cell*, one of the highest-impact journal in biology, began to require graphical abstracts from accepted articles as one of the author guidelines:

> `A graphical abstract should allow readers to quickly gain an understanding of the main take-home message of the paper and is intended to encourage browsing, promote interdisciplinary scholarship, and help readers identify more quickly which papers are most relevant to their research interests.`

Not only *Cell*, publishers such as *Nature*, *Science*, *PLoS* etc. also see the advantage of providing graphical information and thus ask an author to select one figure in her submission as the strike image. This new guideline reveals a visible trend of graphical abstract and aligns with the

motivation of our study. Aimed to this existed demand and application, we use natural language processing techniques for automatically labelling the key figure.

We define the key figure as the figure that best summarizes a paper. Under an assumption that the caption of a key figure likely overlaps a great portion of the abstract, we choose the figure with the caption most similar to the paper abstract to be the key figure. We measure the similarity using the off-the-shelf functions from the python Natural Language Toolkit (NLTK) and gensim [78]. For each paper, we used the stemmed captions and abstract to build a dictionary with one-gram and then transform the text paragraphs into a text corpus. Next we build a TF-IDF model [3] to calculate the cosine similarity [93] for each caption compared to the abstract. The figure with the highest cosine similarity to the abstract is selected as the key figure.

We include the caption similarities together with the figure metadata. As a feature of our figure-centric search engine, we mark the selected key figures for reference (Section 5.1).

3.4 Acquiring Article Influence

To assess the influence of a particular paper, we used the article-level Eigenfactor (ALEF) score developed by our collaborators [46, 99, 100]. ALEF is a modified version of the PageRank algorithm [71]. The algorithm uses random walk on the article-level citation graph, where each vertex is a paper and each directed edge is a citation. Because a random walker will only move backwards in time using the standard PageRank approach, the algorithm is modified to reduce the number of steps the random walker takes and teleports the random walker to links rather than nodes [55, 100]. The ALEF ranking method has been shown to outperform simple citation counts and standard PageRank approaches [99].

We acquire the bibliometrics data from Eigenfactor.org. The dataset consists of the metadata and ALEF scores of 5,538,322 papers in biology, biomedical, and biochemistry area, mainly collected from PubMed. The papers are also clustered using mapequation [80], a hierarchical clustering algorithm that compresses a description of a random walker as a proxy for real flow on a network, finds the shortest multilevel description of the random walker and returns the best hierarchical clustering of the network. The cluster label of a paper reveals its research field in hier-

archy, for instance, a cluster label “1:2:6:24” specifies a research field of “kinases synaptic” under a broader field of ”receptor ion”, represented by “1:2:6”. We use the clusters for recommending relevant figures as a function of our figure-centric search engine, describing in Section 5.3.2.

3.5 Summary

VizioMetrics dataset is based on 4,781,742 images downloaded from PubMed Central. The images are categorized into equations, diagrams, photos, tables, and plots. Together with the bibliometrics data acquired from Eigenfactor.org, the VizioMetrics dataset contains paper metadata, measurement of article influence, machine-labelled figure types, article cluster labels and cosine similarity between captions and abstracts for determining key figures.

Chapter 4

FIGURE ANALYSIS ALGORITHMS

In order to provide faceted search feature on our figure-centric search engine, we developed an image processing pipeline to classify figure images (Figure 4.1) by the figure types. We first download and extract the images in AWS (Amazon Web Services). We then classify each figure as either *multi-chart* or *singleton*. Each figure identified as multi-chart is dismantled into a set of singleton figures. All singleton figures (including those dismantled from multi-chart figures) are labeled with their figure types. In the first-generation pipeline, we classify the figures into five categories:

- Equation (e.g., embedded equations, Greek and Latin characters)
- Diagram (e.g., schematics, conceptual diagrams, flow charts, architecture diagrams, illustrations)
- Photo (e.g., microscopy images, diagnostic images, radiology images, fluorescence imaging)
- Table (any tabular structures with text or numeric data in the cells)
- Plot (e.g., bar charts, scatter plots, line charts)

In the second-generation pipeline, we separate dendrogram (tree graph), metabolic pathway from the diagram and electrophoresis gels from the photo as individual types. These tree figure types are distinct blobs with rich population. Furthermore, they encodes specific information that we are interested to perform further analysis. In the following sections, we will describe the algorithm for each box in Figure 4.1.

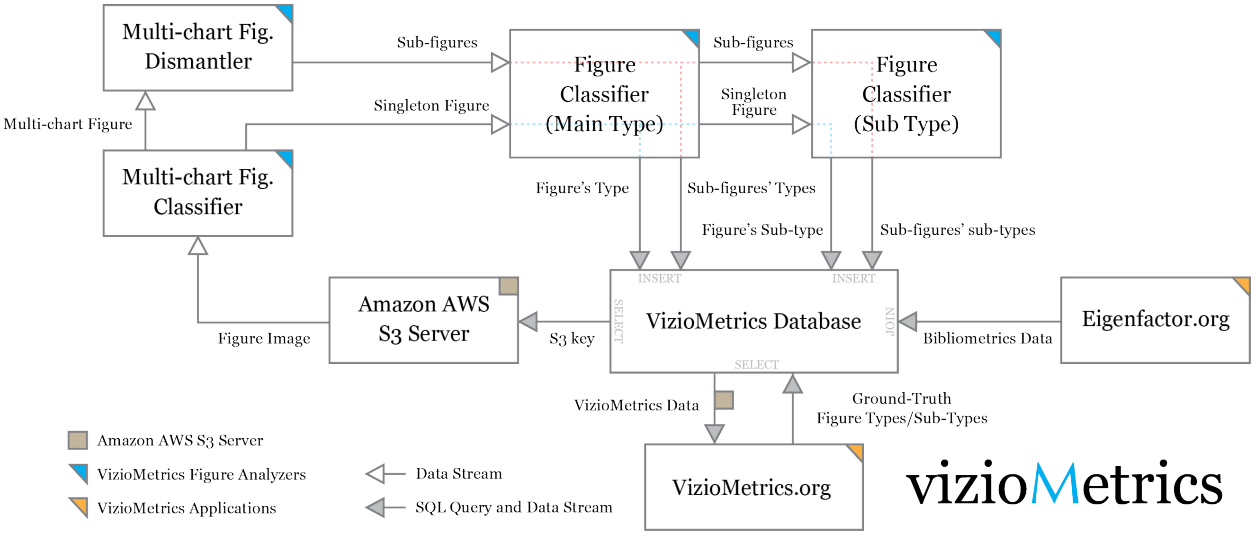

Figure 4.1: VizioMetrics.org system overview. We store the images in Amazon's S3 service. Image paths, figure captions, paper metadata and classification result are stored in the database. The figure analysis system acquires the file keys from the database, downloads the image files, and feeds them into the figure processing pipeline. The final classification results are stored in the database as the sources for the application prototype.

4.1 Figure Dismantling

An unavoidable problem for figure classification is the ubiquitous use of multi-chart figures: single images with multiple embedded sub-plots. Such figures account for approximately 30% of the figures in our corpus. We propose a dismantling algorithm to separate sub-figures from a composite figure. In addition, a part of the algorithm can be used as a feature descriptor for identifying composite figures. We will describe the algorithm of dismantling first and the feature descriptor in the next sub-section. The dismantling algorithm is comprised of three steps: (1) splitting, (2) merging and (3) selecting. In first step, we recursively segment an original image into separate sub-images by analyzing empty space and applying assumptions about layout. In the second step, we use an SVM classifier to distinguish complete sub-figures from auxiliary fragments (ticks, labels, legends, annotations) or empty regions. In the third step, we compare the results produced by alternative initial segmentation strategies by using a scoring function and select the best choice as the final output.

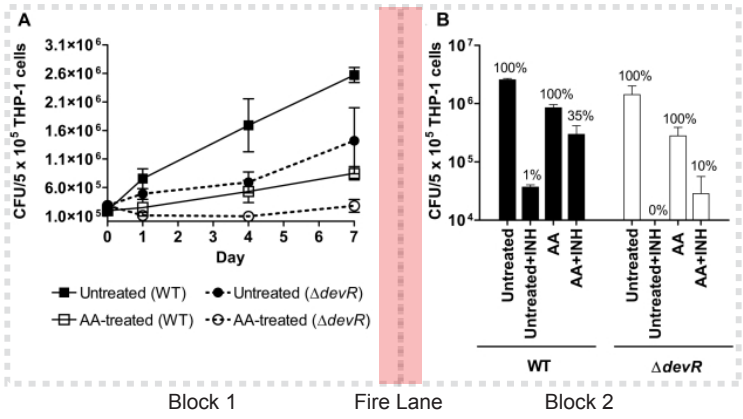

Figure 4.2: A figure containing two charts. Each chart can be covered by one block. A blank stripe in the middle separates them as a "fire lane". (The original figure created by Toriyama et al. [94], used under CC BY 4.0/ Modified from original.)

4.1.1 Step 1: Splitting

The splitting algorithm recursively decomposes the original figure into sub-images. Authors assemble multiple visualizations together in a single figure to accommodate a limited space budget or to relate multiple visualizations into a coherent argument. We made a few observations about how these figures are assembled that guide the design of our splitting algorithm: First, the layout typically involves a hierarchical rectangular subdivision as opposed to an arbitrarily unstructured collage. Second, authors often include a narrow blank buffer between two sub-figures as a "fire lane" to ensure that the overall layout is readable (Figure 4.2). Third, paper-based figures are typically set against a light-colored background. We will discuss figures that violate these assumptions in Section 4.1.4.

Based on these assumptions, our splitting algorithm recursively locates empty spaces and divides the multi-chart figure into *blocks*. Based on our rectangularity assumption, we locate empty spaces by seeking wholly blank rows or columns as opposed to empty pixels. We first convert the color image into grayscale, and then compute a histogram for rows (and a second histogram for columns) by summing the pixel values of each row (or column). Figure 4.3(b) gives an example

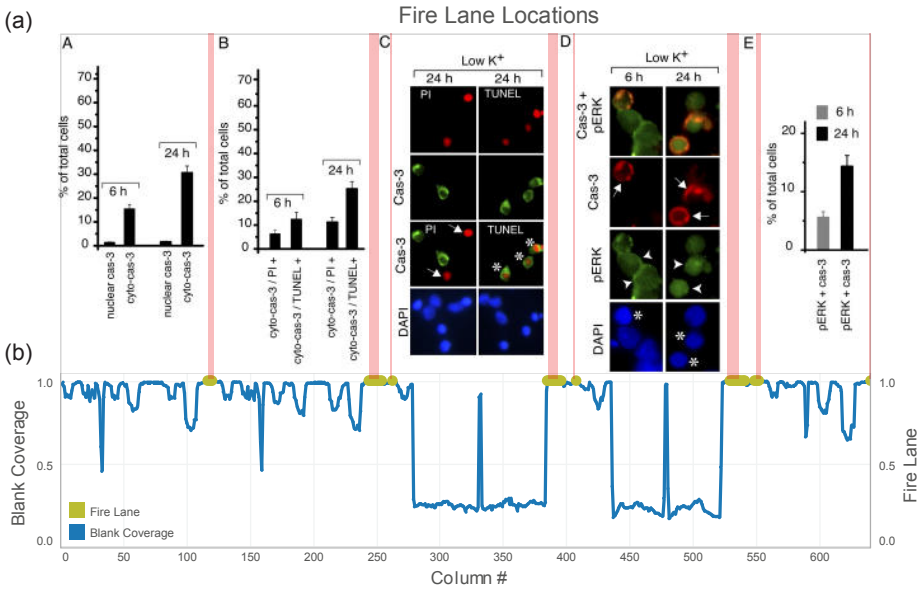

Figure 4.3: (a) Fire lanes. We locate the lanes by using the histogram of columns. Orange dots represent qualified columns that pass the thresholds. (b) Histogram of columns. (Original figures created by ©Subramaniam et al. [89], 2004. Originally published in *The Journal of cell biology*. http://doi.org/10.1083/jcb.200403028.)

of a figure with its corresponding histogram for the columns. Candidate fire lanes appear as peaks or plateaus in the histogram with a value near the maximum, so we normalize the histogram to its maximum value and apply a high-pass empty threshold θ_e to obtain a candidate set of "blank" rows (or columns). The maximum value does not necessarily indicate a blank row or column, because there may be no entirely blank rows or columns. For example, the green vertical line in Figure 4.4(a) is the maximum pixel value sum, but is not blank and is not a good choice as a fire lane. To address this issue, we apply a low-pass variance threshold θ_{var1} to filter such items by their relatively high variances (Figure 4.4(b)). We use a second method to detect empty spaces by applying another, stricter low-pass variance threshold θ_{var2} on rows or columns. The first method provides a wider pass window and the second method is well-suited to handle figures with a dark background.

To set the values of the three thresholds, we collected 90 composite figures (avoiding figures with photographs) and ran the splitting step with different combinations of thresholds against this

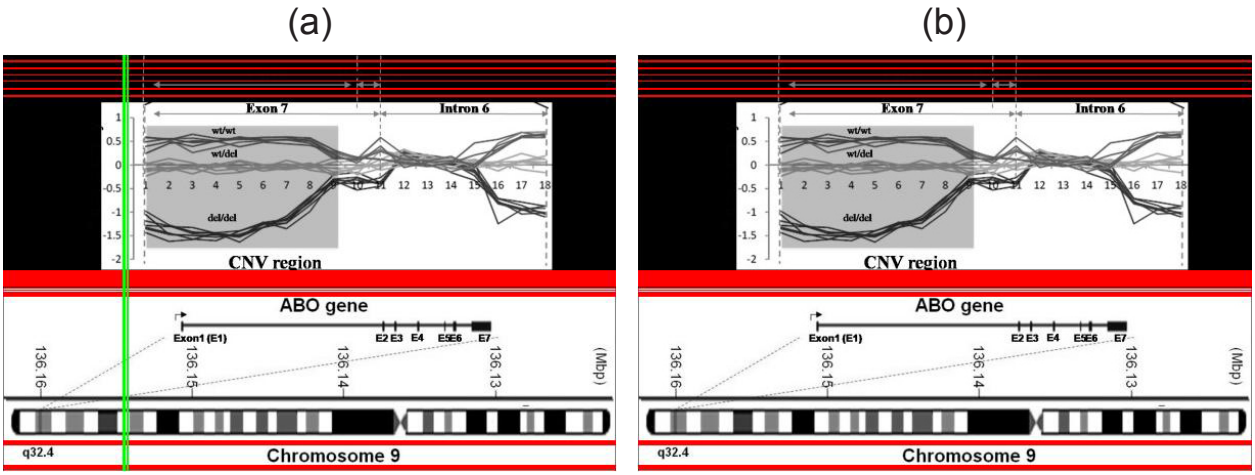

Figure 4.4: The identification of fire lanes is non-trivial. (a) Locating fire lanes without applying the variance threshold θ_{var1} leads to an error: Since there are no entirely blank columns, the maximum value (highlighted in green) is not a qualified fire lane. (b) The disqualified column is filtered by applying $\theta_{var1} = 100$. (The original figure created by Hong et al. [42], used under CC BY 4.0/ Modified from original.)

training set. Since our goal is just to tune these parameters, we make a simplifying assumption that finding the correct number of sub-images implies a perfect split; that is, if the number of divided sub-images equals the correct number of sub-figures determined manually, we assume the division was perfect. The reason for this simplifying assumption is to improve automation for repeated experiments; we did not take the time to manually extract perfect splits for each image with which to compare. Under this analysis, the values for the thresholds that produced the best results were $\theta_e = 0.999$, $\theta_{var1} = 100$, and $\theta_{var2} = 3$.

We group neighboring empty pixel-rows or empty pixel-columns to create empty "fire lanes" as shown in Figure 4.3(a). The width of the fire lane is used in the merge step to determine each sub-image's nearest neighbor. Half of each fire lane is assigned to each of the two blocks; each block becomes a new image input to be analyzed recursively. Row-oriented splits and column-oriented splits are alternatively performed, recursively, until no fire lane is found within a block. The recursion occurs at least two times to ensure both orientations are computed at least once.

Different initial splitting orientations can result in different final divisions, so the splitting al-

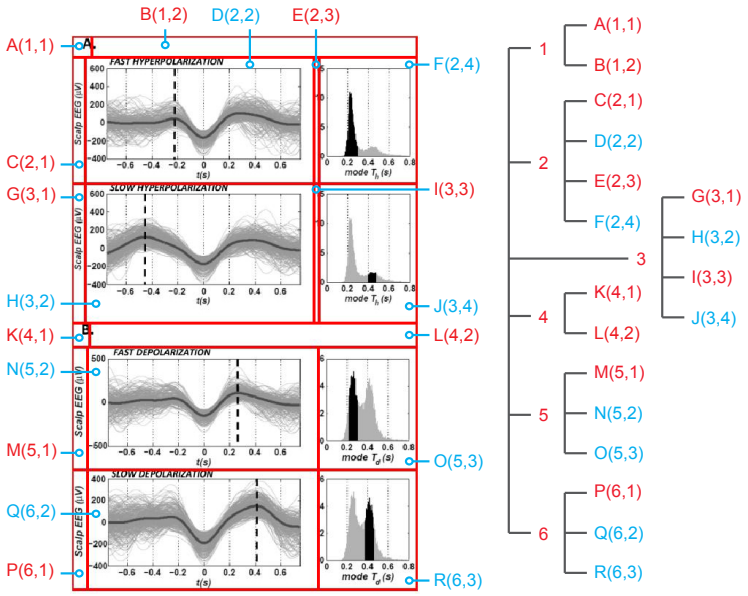

Figure 4.5: The tree structure of a decomposition. This multi-chart image was split using column-orientation. The result of the splitting step can form a tree structure. The numbers in parentheses present the sections in each splitting level that the block belongs to. For instance, H(3, 2) refers that block H is in the third section when the original multi-chart figure is split. Then it is the second sub-section when the third section is split again. With the assistance of the classifier, we color standalone blocks by blue and auxiliary blocks by red. (The original figure created by Botella-Soler et al. [11], used under CC BY 4.0/ Modified from original.)

gorithm is performed twice: once beginning vertically and once beginning horizontally. We individually execute merging for the two results and automatically evaluate the merging results in step 3. The split with higher score is taken as the final decomposition.

4.1.2 Step 2: Merging

The merging algorithm receives the splitting result as input and then proceeds in two substeps: First, we use an SVM-based classifier to distinguish *standalone* sub-figures representing meaningful visualizations from *auxiliary* blocks that are only present as annotations for one or more standalone sub-figures. Second, we recursively merge auxiliary blocks, assigning each to its nearest block, until all auxiliary blocks are associated with one (or more) standalone sub-figures. We

refer to this process as hierarchical merging. If two neighboring blocks have incongruent edges, a non-convex shape may result. In this case, we perform *T-merging*: we search nearby for sub-figures that can fill the non-convexity in the shape. We will discuss the details of the classifier, Hierarchical Merging, and T-Merging in this section.

Training the SVM-based Binary Classifier

Figure 4.5 shows an example of an intermediate state while merging, consisting of 18 sub-images from the composite figure. Sub-images labeled (D, F, H, J, N, O, Q, R) are classified as standalone blocks. All others are classified as auxiliary blocks. The goal of the merging algorithm is to remove auxiliary blocks by assigning them to one or more standalone blocks. To recognize auxiliary blocks, we extract a set of features for each block and train an SVM-based classifier. The features selected are based on the assumption that the authors tend to follow implicit rules about balancing image dimensions and distributing empty space within each figure, and that these rules are violated for auxiliary annotations. To describe the dimensions of the block, we compute proportional area, height and width relative to that of the original image, as well as the aspect ratio. To describe the distribution of empty space, we use the same thresholds from the splitting step to locate entirely blank rows or columns and then compute the proportion of the total area covered by the pixels of these blank elements. We do not consider the overall proportion of empty pixels, because many visualizations use an empty background — consider a scatter plot, where the only non-empty pixels are the axes, the labels, the legend, and the glyphs. As a result, blank rows and columns should be penalized, but blank pixels should not necessarily be penalized.

Blank coverage alone does not sufficiently penalize sub-figures that have large blocks of contiguous empty space; a pattern we see frequently in auxiliary sub-images. For example, an auxiliary legend offset in the upper right corner of a figure will have large contiguous blocks of white space below and to the left of it. To describe these cases where empty space tends to be concentrated in particular areas, we divide each sub-image into k equal-size sections via horizontal cuts and another k sections via vertical cuts. We then extract one feature for each horizontal and vertical section; $2k$ features total. Each feature f_i is computed as the proportion of blank rows in section

Table 4.1: The features used to classify sub-images as either standalone sub-figures or auxiliary fragments. We used $k = 5$ for our experiment; thus the feature vector consists of 15 elements. We achieved classification accuracy of 98.1%, suggesting that these geometric and whitespace-oriented features well describe the differences between the two categories.

Area Ratio = $\frac{Area_{sub\text{-}image}}{Area_{composite\ figure}}$
Height Ratio = $\frac{Height_{sub\text{-}image}}{Height_{composite\ figure}}$
Width Ratio = $\frac{Width_{sub\text{-}image}}{Width_{composite\ figure}}$
Aspect Ratio = $\frac{Hight_{sub\text{-}image}}{Width_{sub\text{-}image}}$
Blank Coverage = $\frac{\text{sum of pixels in blank rows and columns}}{\text{total number of pixels}}$
for i 0..k: Percent blank rows in horiz. section i = $\frac{\text{num. of blank rows in } i}{Height_i}$
for j 0..k: Percent blank columns in vert. section j = $\frac{\text{num. of blank columns in } j}{Width_j}$

i. To determine a suitable k, we experimented with different values from $k = 0$ to $k = 10$ on the training data and set $k = 5$ based on the results. In this paper, we do not consider further optimizing this parameter. With the combination of the dimensional features and the empty-space features, we obtain a 15-element feature vector for each sub-image. These features are summarized in Table 4.1.

As an example of how these features manifest in practice, consider Figure 4.6. This image has 26.6% blank coverage; blank columns are colored green and blank rows are colored red. As with most visualizations in the literature, the overall percentage of blank pixels is very high, but the percentage of blank rows and columns is relatively low. We divide the image into horizontal and

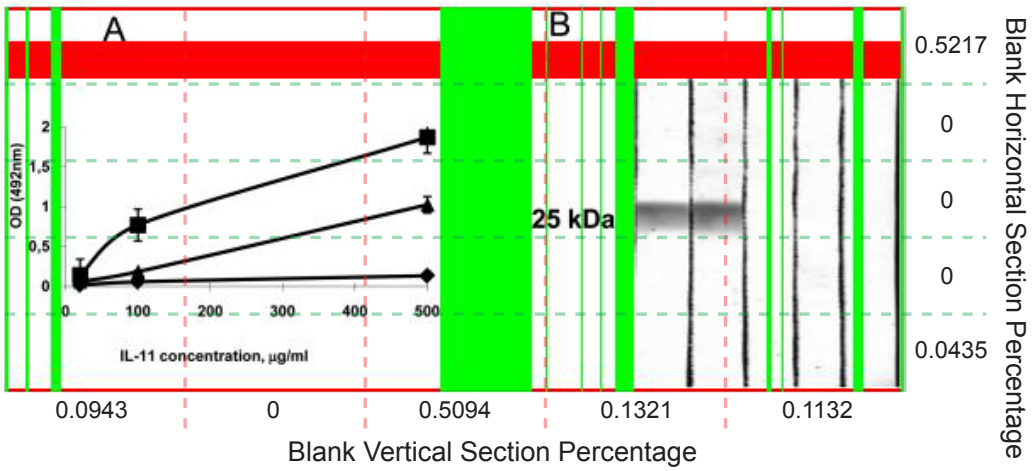

Figure 4.6: Blank coverage according to blank rows (red) and blank columns (green). We divided the image into 5 sections horizontally and vertically. In each section, we computed the percentage of blank-row or blank-column respectively. The 10 vectors form a portion of image feature. (The original figure created by Kapina et al. [50], used under CC BY 4.0/ Modified from original.)

vertical sections as indicated by the green and red dashed lines. The decimals indicate percentages of blank row or column of their nearby sections. The complete 15-element feature vector of this image is {1, 1, 1, 0.4272, 0.2656, 0.5217, 0, 0, 0, 0.0435, 0.0943, 0, 0.5094, 0.1321, 0.1132}.

To evaluate our classifier, we collected another corpus containing 213 composite figures from the same source for training independence. The splitting algorithm was used to produce 7541 sub-images from the corpus. For evaluation, we manually classified a set of 6524 standalone sub-images and 1017 auxiliary sub-images. We used LibSVM [15] and set all parameters as default to train the model. Table 4.2 shows the performance of the image features for merging determination by training an SVM-based binary classifier through 10-fold cross-validation on the sub-image set. The classification accuracy is 98.1%.

Hierarchical Merging

The result of the splitting step forms a tree structure as shown in Figure 4.5. Merging starts from the leaves of the tree. All leaves can only merge with other leaves in the same level. After completing

Table 4.2: Classification accuracy calculated by 10-fold cross-validation

Class	Correct	Incorrect	Accuracy (%)
Auxiliary	6482	42	99.4
Standalone	917	100	90.2
Total	7399	142	98.1

all possible merges among siblings, we transfer the newly merged block to their parents' level as new leaves. Hierarchical merging stops after it finishes merging in the top level.

In each level, we re-run the classifier to determine auxiliary blocks. We then induce a function on the set of blocks, assigning each block to its nearest adjacent neighbor called its *merge target*. A block and its merge target are called a merge pair. Under the assumption that the width of a fire lane indicates the strength of relationship between the two blocks, the merge target for a block is the adjacent neighbor with the narrowest lane between them. Figure 4.7(a) shows a combination of two blocks. The new block is the smallest rectangle that covers both merging blocks. Only adjacent blocks are allowed to merge together. If there are two or more qualified blocks, we break the tie by the shortest distance between centroids of blocks.

If after merging all auxiliary blocks at one level of the tree we find that the resulting shape is non-rectangular, we attempt to apply T-merging or pass the result on to the next higher level. For example, Figures 4.7(b) and 4.7(c) can be merged since the result is rectangular. Figures 4.7(d) and 4.7(e) are forbidden. In this case, the merging of the block pair is skipped and the auxiliary block is labeled as standalone and processed using T-Merging, described next. We repeat the local merging until the statuses of all blocks are standalone. We then pass these blocks up to the next level, reclassify them, and repeat the local merging again.

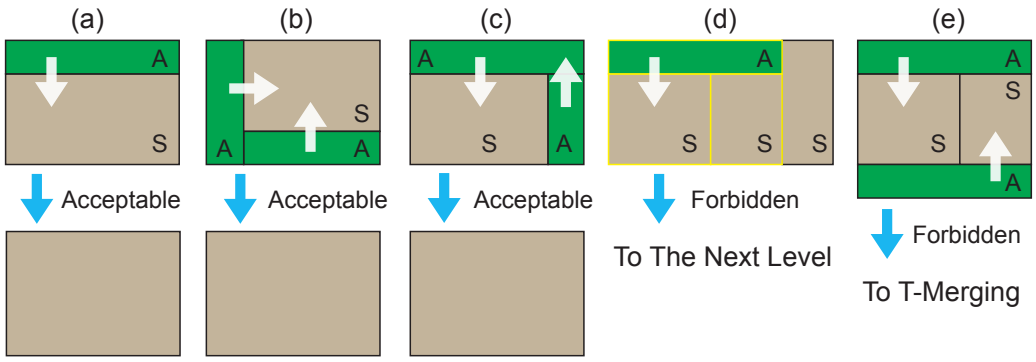

Figure 4.7: Examples of Hierarchical Merging. In all cases, the goal is to merge all auxiliary blocks, (labeled A) into standalone blocks (labeled S). Each merge operation is indicated by a white arrow. (a) An acceptable merge. The new block is the smallest rectangle that covers both merging blocks. (b) Two different merge paths that lead to the same result. (c) Another case of acceptable multi-merging. (d) This merging is forbidden because after merging the auxiliary into the standalone block the resulting shape is non-rectangular. The operation only involves the blocks with yellow outline. Once the local merging in this level is completed, it repeats again in the next level, which will involve the very right standalone block. (e) Another case of forbidden merging because of the same reason. After completing Hierarchical Merging, the residual auxiliary blocks will be handled by T-Merging.

T-Merging

T-Merging handles residual auxiliary blocks ignored in Hierarchical Merging. These are usually shared titles, shared axes or text annotations that apply to multiple sub-figures; e.g., Figure 4.7(d) and Figure 4.7(e). As shown in Figure 4.8, merging the auxiliary block 1 ("the legacy"), to any adjacent standalone block generates a non-rectangular shape. We define block 2 and block 3 as legatees[1] to proportionally share block 1. We find the set of legatees by the following procedure: For each edge e of the legacy, find all blocks that share any part of e and construct a set. If merging this set as a unit produces a rectangle, then it is a qualified set. If multiple edges produce qualified sets, choose the edge with t the narrowest fire lane. In Figure 4.8, only the set consisting of block 2 and 3 satisfies the above criteria; blocks 4, 5, 6 are not proper legatees. Figure 4.9 illustrates the evolution from a source image through Hierarchical Merging to its T-merged output.

[1] meaning "those who will receive the legacy"

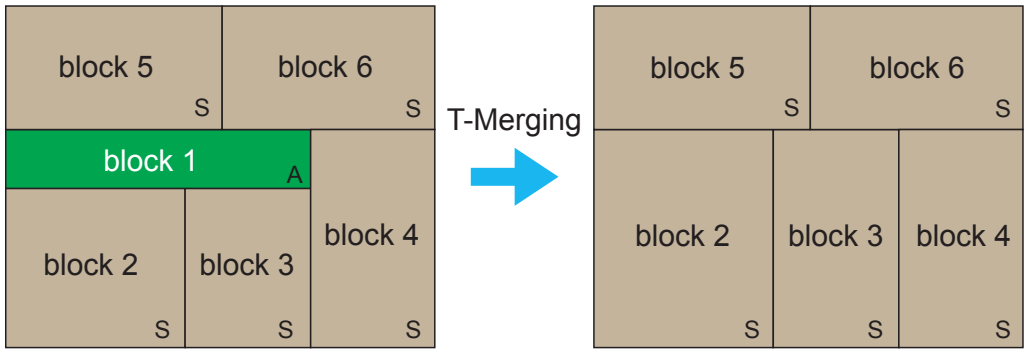

Figure 4.8: Examples of T-Merging. The legacy (block 1) is marked by white color in text. According to our algorithm, only block 2 and block 3 are qualified to share block 1.

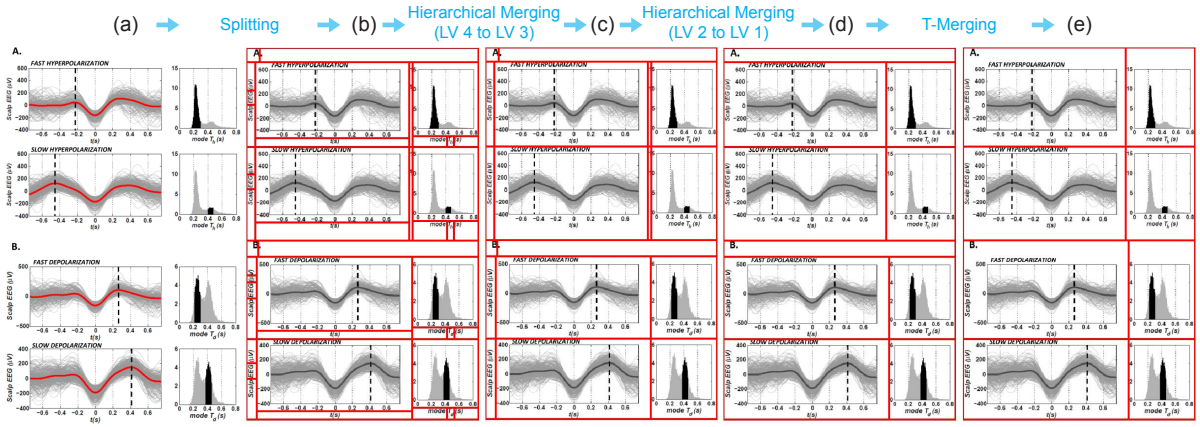

Figure 4.9: (a) Composite figure. (b) Splitting result. (c) Intermediate state of Hierarchical Merging after completing level 3. (d) Hierarchical Merging result. The very-top block and the middle block require T-Merging. (e) T-Merging result. (The original figure created by Botella-Soler et al. [11], used under CC BY 4.0/ Modified from original.)

4.1.3 Step 3: Selecting

The splitting and merging steps may produce different results from different initial splitting orientations (Figure 4.10). Step 3 scores the two different results and selects the one with higher score as the final output. Under the assumption that authors tend to follow implicit rules about balancing image dimensions, the decomposition that produces more sub-images with similar dimensions is given a higher score. To capture this intuition, we define the scoring function as

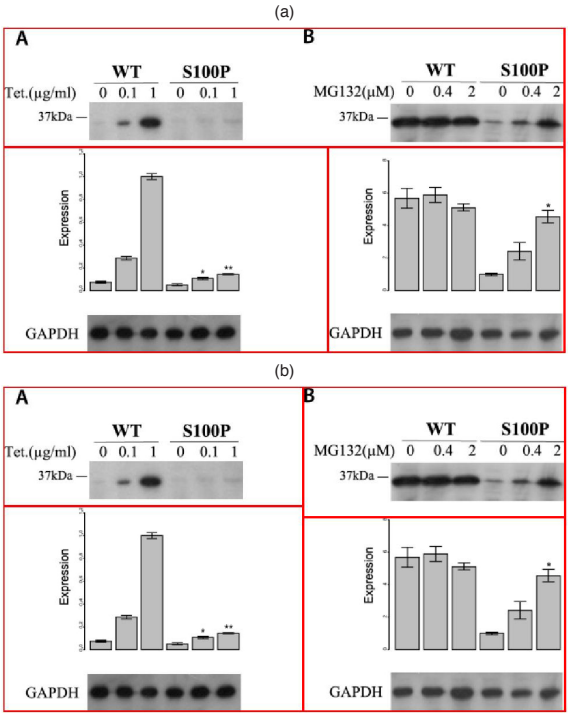

Figure 4.10: Results of different initial splitting orientations. (a) Split begins from horizontal (initially row-oriented), and a lower score due to mismatched elements. (b) Split begins from vertical (initially column-oriented), and a higher score. (The original figure created by Türumen et al. [95], used under CC BY 4.0/ Modified from original.)

$$S_{decomposition} = 4 \sum_{i \in \text{blocks}} \sqrt{A_i} - 2\alpha \sum_{i,j \in \text{Pairs}} |l_i^{top} - l_j^{top}| + |l_i^{left} - l_j^{left}|,$$

where A_i is the area of the corresponding block, α is a penalty coefficient and l_i^{top} is the length of the top edge of block i (respectively, $left$). Each element of the set *Pairs* is a pair of blocks i, j, where j is the block that has the most similar dimensions to i for $i \neq j$. The two coefficients normalize the two terms to the full perimeter. The formula enforces a geometric property of composite figures: The first term obtains its maximum value when all blocks are equal in size. The second term subtracts the difference between each block and its most similar neighbor to reward repeating patterns and penalize diversity in the set. The penalty coefficient weights the importance of dimensional difference. We assigned $\alpha = 1$ in our experiment.

4.1.4 Experimental Result

In this section, we describe experiments designed to answer the following questions: a) Can our algorithm be used to estimate visualization diversity, a weaker quality metric sufficient for many of our target applications? (Yes; Table 4.3) b) Can our algorithm effectively extract correct sub-figures, a stronger quality metric? (Yes; Table 4.3) c) Could a simpler method work just as well as our algorithm? (No; Table 4.3) d) Is step 3 of the algorithm (selection) necessary and effective? (Yes; Figure 4.11) e) Where does the algorithm make mistakes? (Figure 4.14)

The corpus we used for our experiments was randomly collected from PubMed image corpus. We manually identified the multi-chart figures and divided them into a testing set and a training set. We trained the classifier and performed cross-evaluation with the training set, reserving the test set for a final experimental evaluation. The testing set S for the experiments contains 261 multi-chart figures related to biology, biomedicine, or biochemistry. Each figure contains at least two different types of visualizations; e.g., a line plot and a scatter plot, a photograph and a bar chart, etc, defined as "composite figure". We ignored multi-chart figures comprised of single-type figures in this experiment for the convenience of evaluation, described later in the first question. We evaluated performance in two ways: (1) type-based evaluation, a simpler metric in which we attempt to count the number of distinct types of visualizations within a single figure, and (2) chart-based evaluation, a stronger metric in which we attempt to perfectly recover all sub-figures within a composite figure. We implemented the algorithm in Matlab and conducted the experiments based on this version. We re-implemented the dismantling algorithm in Python 2.7 for the practical use in the figure processing pipeline.

Can our algorithm be used to estimate visualization diversity? The motivation for type-based evaluation is that some of our target applications in bibliometrics and search services need only know the presence or absence of particular types of visualizations in each figure to afford improved search or to collect aggregate statistics — it is not always required to precisely extract a perfect sub-figure, as long as we can tell what type of figure it is. For example, the presence or absence of an electrophoresis gel image appears to be a strong predictor of whether the paper is in the area of

experimental molecular biology; we need not differentiate between a sub-figure with one gel and a sub-figure with several gels. Moreover, it is not always obvious what the correct answer should be when decomposing collections of sub-figures of homogeneous type: Part of Figure 4.14(e) contains a number of repeated small multiples of the same type — it is not clear that the correct answer is to subdivide all of these individually. Intuitively, we are assessing the algorithms' ability to eliminate ambiguity about what types of visualizations are being employed by a given figure, since this task is a primitive in many of our target applications.

To perform type-based evaluations we label the test set by manually counting the number of distinct visualization types in each composite figure. For example, Figure 4.2 has two types of visualizations, a line chart and a bar chart; Figure 4.5 also has two types of visualizations, a line chart and an area chart; Figure 4.10(a) also has two types of visualizations, bar charts and electrophoresis gels. We then run the decomposition algorithm and manually distinguish correct extractions from incorrect extractions. Only *homogeneous* sub-images — those containing only one type of visualization — are considered correct. For example, the top block in Figure 4.10(a) is considered correct, because both sub-figures are the same type of visualization: an electrophoresis gel image. The bottom two blocks of Figure 4.10(a) are considered incorrect, since each contains both a bar chart and a gel.

Using only the homogeneous sub-images (the heterogeneous sub-images are considered incorrect), we manually count the number of distinct visualization types found for each figure. We compare this number with the number of distinct visualization types found by manual inspection of the original figure. For example, in Figure 4.10(a), the algorithm produced one homogeneous sub-image (the top portion), so only one visualization type was discovered. However, the original image has two distinct visualization types. So our result for this figure would be 50%.

To determine the overall accuracy we define a function $diversity : Figure \rightarrow Int$ as $diversity(f) = |\{type(s) \quad | \quad s \in decompose(f)\}|$, where *decompose* returns the set of subfigures and *type* classifies each subfigure as a scatterplot, line plot, etc. The final return value is the number of distinct types that appear in the figure. We then sum the diversity scores for all figures in the corpus. We compute this value twice: once using our automatic version of the *decompose* function and once

using a manual process. Finally, we divide the total diversity computed automatically by the total diversity computed manually to determine the overall quality metric. The automatic method is not generally capable of finding more types than are present in the figure, so this metric is bounded above by 1. In our experiment, we obtained the diversity score of 591 and 640 respectively from automatic decomposition and manual process. The accuracy by this metric is therefore 92.3%.

Table 4.3: Chart-based evaluation. Where S_{all} denotes the entire composite figure set, $S_{p\leq 8}$ denotes the subset of composite figures containing eight or fewer sub-figures, and $S_{p>8}$ denotes the subset of composite figures containing nine or more sub-figures. We compared our main approach to a splitting-only method based on our splitting algorithm. The recall and the precision of correct sub-images, as well as the accuracy of decomposition were significantly enhanced. Our technique achieved a better performance for a subset of composite figures that contains eight or fewer sub-figures.

		Recall of Correct Sub-images	Precision of Correct Sub-images	Accuracy of Perfect Decomposition
Splitting-only Approach	S_{all}	54.3% (833/1534)	53.1% (833/1569)	16.1% (42/261)
Main Approach	S_{all}	67.5% (1035/1534)	80.8% (1035/1281)	57.9% (151/261)
	$S_{p\leq 8}$	80.1% (811/1002)	85.1% (811/953)	67.1% (149/222)
	$S_{p>8}$	42.1% (224/532)	68.3% (224/328)	5.1% (2/39)

Can our algorithm effectively extract correct sub-figures? For chart-based evaluation, we attempt to perfectly extract the exact subfigures found by manual inspection, and measure precision and recall. For instance, Figure 4.3, Figure 4.5, and Figure 4.10(b) contain 5, 8, and 6 sub-figures respectively. To obtain ground truth, we manually extracted 1534 visualizations from the entire

image set *S*; about 5.88 visualizations per composite figure on average. In this experiment, a sub-image that includes exactly one visualization is defined as correctly extracted; exceptions are described next. However, a sub-image that crops a portion of a visualization (e.g., Figure 4.14(a), bottom), or includes only auxiliary annotations (e.g., Figure 4.14(b), bottom right), or includes two or more visualizations (e.g., Figure 4.14(b), top left) is considered incorrect. These criteria are stricter than necessary for many applications of the algorithm; for example, partial visualizations or visualizations with sub-structure will often still be properly recognized by a visualization-type classifier and can therefore be used for analysis. However, this metric provides a reasonable lower bound on quality.

We make an exception to these criteria: We consider an array of photographic images to be one unit if the author assigns a part label for the array. This exception is to ensure that we do not artificially improve our results: The algorithm is very effective at decomposing arrays of photos, but it is not obvious that these arrays should always be decomposed; the set is often treated as a unit. In this analysis, we also ignore cases where an auxiliary annotation is incorrectly assigned to one owner instead of another. The reason is that we find a number of ambiguous cases where "ownership" of an auxiliary annotation is not well-defined.

We define a notion of recall and precision based on these correctness criteria. To compute recall, the number of correct sub-images returned by the algorithm is divided by the number of correct sub-figures manually counted in the corpus using the same criteria for correctness. To compute precision, we divide the number of correct sub-images returned by the algorithm by the total number of extracted sub-images. Our algorithm achieves recall of 67.5% and precision of 80.8%. In addition, the percentage of figures that are perfectly decomposed — the right number of correct images and no incorrect images — is 57.9%. Table 4.3 summarizes the chart-based evaluation in more detail. Later in this section we will analyze the mistakes made by the algorithm.

Does a simpler method work just as well? For comparison, we measured the performance of our algorithm relative to a simpler split-based algorithm. Here, we modified our splitting step (Section 4.1.1) to make it more viable as a complete algorithm. As presented, our splitting step

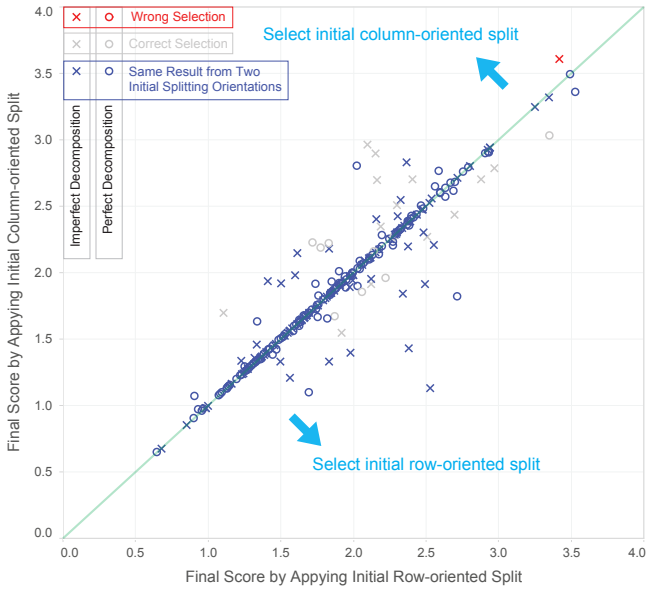

Figure 4.11: Step 3 of the algorithm, selection, makes correct decisions. Circles represent perfectly decomposed figures and crosses represent imperfectly decomposed figures. This scatter plot illustrates that figures with perfect decomposition mostly distribute near the line of slope 1, indicating similar solutions were found by our decomposition algorithm regardless of the starting orientation. The selection step deals with the the grey plots and the red plot, which are composite figures that have different outputs from the two initial splitting orientations. Only one mistaken selection was made and 95.9% accuracy was achieved.

may produce a large number of auxiliary fragments that need to be merged (e.g., Figure 4.9(b)). But a reasonable approach would be to cap the number of recursive steps and see if we could avoid the need to merge altogether. We use two recursive steps — once for vertical and once for horizontal. Also, as a heuristic to try and improve the results, we discarded fire lanes with width less than 4 pixels for the same purpose because most lanes between auxiliary fragments or between auxiliary fragments and effective sub-figures are relatively narrow.

Our results show that this splitting-only algorithm extracted 833 correct sub-images and achieved 54.3% recall and 53.1% precision. Only 16.1% of the original composite figures were decomposed perfectly into exact sub-figures without any errors. By both measures, this simpler method performs significantly worse despite optimizations (Table 4.3).

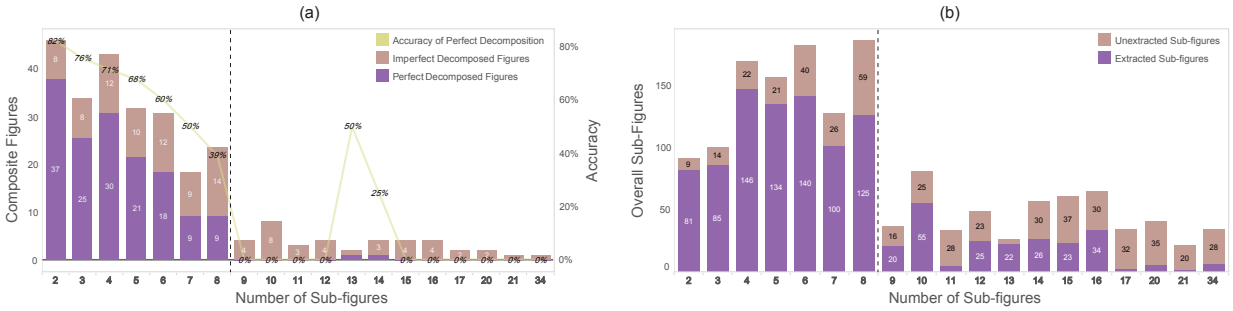

Figure 4.12: (a) Histogram of perfectly decomposed and imperfectly decomposed figures. (b) Histogram of extracted sub-figures. Our decomposition algorithm performed better for composite figures with lower number of sub-figures. Entanglement and over-merging are common issues for images of densely packed sub-figures.

Is step 3 of the algorithm (selection) useful and effective? To evaluate the utility of our selection step, we manually compared the two outputs of different splitting orientations before our algorithm automatically chose one. There are 237 figures that have the same results from the two initial splitting orientations. For the remaining 24 figures that require selecting algorithm, our selection algorithm correctly chose the better output for 23 figures, 11 from initial column-oriented split and 13 from initial row-oriented split. Figure 4.11 shows an overview of all selection scores as computed by the formula in Section 4.1.3. Each point denotes a composite figure. Circles are figures decomposed perfectly and crosses are figures decomposed imperfectly. Figures with perfect decomposition mostly appear near the line of slope 1, indicating that our decomposition algorithm often finds similar solutions for regardless of the starting orientation. However, for points where one score is different than the other, we conclude that the selection step plays an important role.

Where does the algorithm make mistakes? To understand the algorithm's performance more deeply, we considered whether the complexity of the initial figure had any effect on the measured performance. Figure 4.12(a) shows a histogram of composite figures, where each category is a different number of sub-figures. The dark portion of each bar indicates the proportion of composite figures that were perfectly decomposed. The curve, which shows the accuracy of perfect decomposition, decays significantly as the number of sub-figures increases; the algorithm tends to

Table 4.4: Causes of decomposition errors. We categorized errors into three types and seven causes. An imperfect decomposition can regard two or more causes.

	Causes of	Units
Merging errors	Over-merging	46
	Orphan fragment	27
	Mistaken merging target	11
Splitting errors	Entanglement	28
	Noisy fire lane	10
	Insider	1
Selecting errors	Mistaken selecting	1

perform significantly better on figures with eight or fewer sub-figures.

Figure 4.12(b) is a histogram of the total number of sub-figures extracted from each category, regardless of whether or not the entire figure was perfectly decomposed. The black dotted line divides the categories into two subsets. The right subset, comprising composite figures containing nine or more sub-figures, includes only 15.0% source figures but contributes 61.7% of the unextracted sub-figures (i.e., the figures the algorithm failed to properly extract). Thus, a relative low recall of 42.1% was obtained in this subset (Table 4.3). In the left subset, comprising 222 composite figures with eight or fewer sub-figures, the recall was greatly increased to 80.1% (Table 4.3). The two bar charts both show a better performance on composite figures with lower sub-figure populations.

Table 4.4 summarizes causes of decomposed errors. Merging errors include over-merging (47), orphan fragment (27), and mistaken merging target (11). Over-merging means that the algorithm mistakenly merges two or more sub-figures together, and orphan fragment implies a missed opportunity to merge a block containing only auxiliary annotations to it's nearby sub-figure. The two causes are both due to the misclassification by our classifier. Mistaken merging target mostly occurs with tables, diagrams and photo arrays with inner separation wider than nearby fire lanes. Splitting errors are categorized into entanglement (28), noisy fire lane (10), and insider (1). An entangled figure violates our basic assumption of a grid-based layout. Figure 4.14(c) shows an

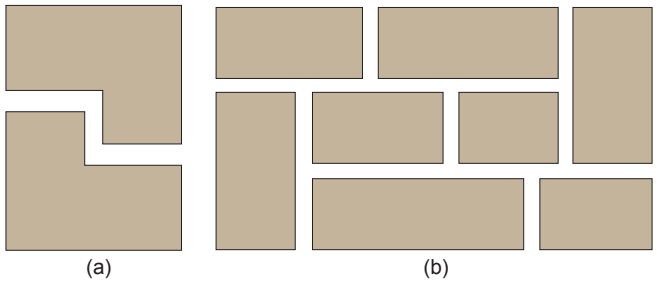

Figure 4.13: Cases for which the splitting algorithm is not appropriate. (a) Irregular outer bound of sub-figures may form a zigzag fire lane. (b) There is no end-to-end fire lane.

example of entanglement. Noise made by image compression in empty space causes the same issue. Embedded charts are also not extractable by our algorithm. Figure 4.14 shows a collection of imperfectly decomposed examples.

The current decomposition algorithm is suitable for grid-aligned multi-chart visualizations, where there exists at least one edge-to-edge fire lane that can bootstrap the process. Figure 4.13 shows two examples that do not satisfy this criterion, and for which our algorithm does not produce a result. Our algorithm is also ill-suited for arrays of similar sub-figures for which it is ambiguous and subjective whether or not they should be considered as one coherent unit. We chose to maximally penalize our algorithm by assuming that every individual element should be considered a separate sub-figure.

4.2 Multi-chart Figure Classification

Attempting to dismantle every figure image in our corpus would be prohibitively expensive and extremely wasteful; only about 30% of the images are multi-chart figures. We therefore developed a simple and fast pre-classifier to distinguish multi-chart figures from singleton figures in order to reduce the number of dismantled singletons.

We designed the method based on two observations: that multi-chart figures tend to have a different size and shape than singleton figures, and that the layout of a multi-chart figure tends to follow a regular grid pattern. Based on these two observations, we constructed a feature vector

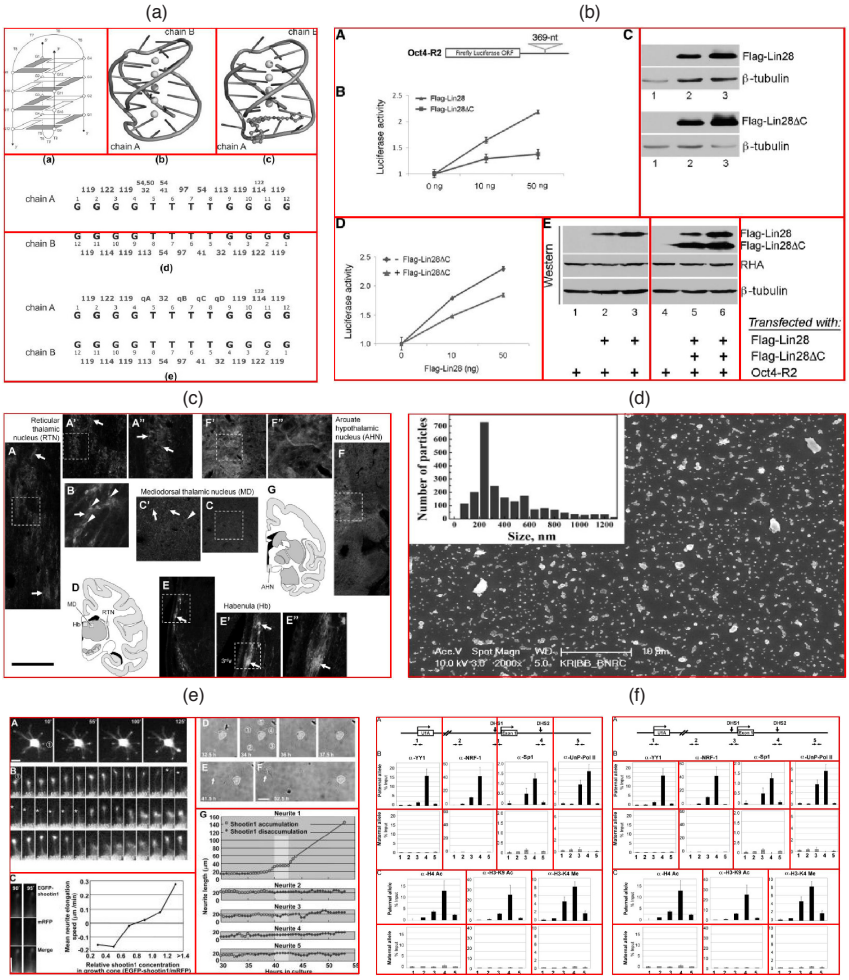

Figure 4.14: Decomposition errors. (a) Mistaken merging target. The chain B in chart (d*) mistakenly merged to the chart (e*) due to an inner wider stripe. (b) Over-merging and orphan fragment. Diagram A was misclassified to be auxiliary and caused a merging error. Furthermore, the label of electrophoresis gels are separated. (c) Entanglement. (d) Insider. (e) Noisy between the photo arrays is generated during image compression. (f) Our algorithm mistakenly chose the result derived by initial column-oriented split (left) in selecting step. (* denotes original marks of assignment in the source image.) (Original figures created by Čech et al. [96], Neddens and Buonanno [68], Gongalsky et al. [34], used under CC BY 4.0/ Modified from original. The original figures are created by ©Toriyama et al. [94], 2006. Originally published in *The Journal of cell biology*. http://doi.org/10.1083/jcb.200604160. The original figures created by Jin et al. [48], Rodriguez-Jato et al. [79]), published in *Nucleic Acids Research*. http://doi.org/10.1093/nar/gkq1350 and http://doi.org/10.1093/nar/gki786 respectively.)

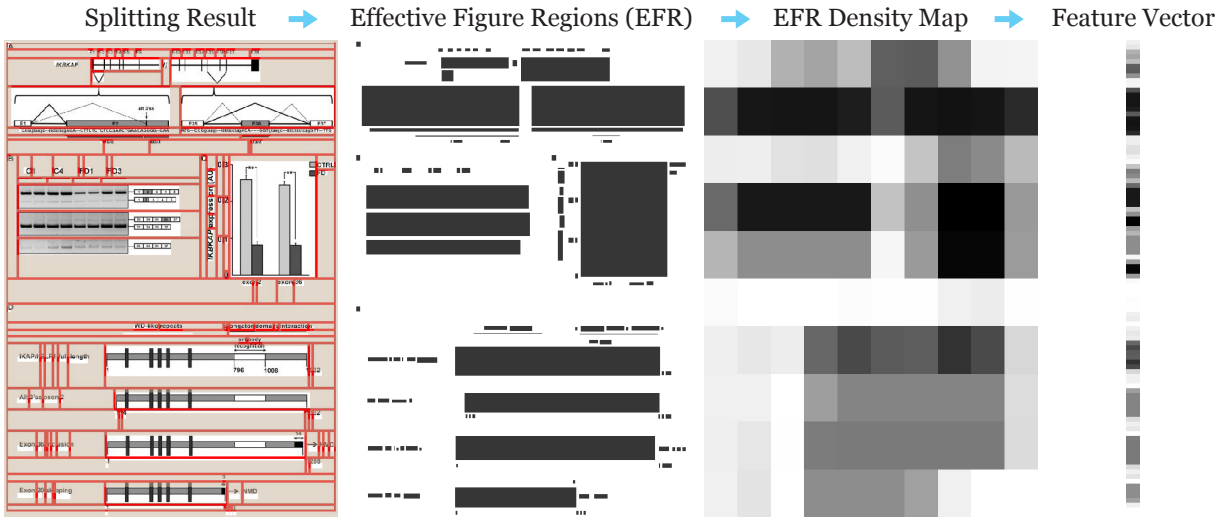

Figure 4.15: Recognizing mulfti-chart images. After splitting the figure into distinct blocks, the dismantling algorithm marks the effective figure regions (EFR) then downsamples the EFR into $n \times n$ blocks that form a $n^2 \times 1$ feature vector. These vectors are used to train the classifier. (Original figures created by Nathalie Boone et al. [10], used under CC BY 4.0/ Modified from original.)

with K (K = M + N) elements: M elements based on the size and shape, and N elements based on the grid layout. The M elements consist of the image height ratio($height_i$ / $height_{avg}$) and the image width ratio($width_i$ / $width_{avg}$) where the denominators are average image height and average image width of all images in the training set respectively. The N elements are derived from the output of splitting algorithm of the dismantler.

Figure 4.15 shows the splitting, and the red lines indicate the boundaries between fragments. For each block, we mark the minimal rectangular region that contains non-empty pixels, so that we can obtain the effective figure regions (EFR) and use them as a mask. We subdivide the mask into $n \times n$ blocks and compute the proportion of EFR in each block as defined as the EFR density map. Finally, we squeeze the values into a 1-D vector with n^2 elements.

4.2.1 Experimental Result

We implemented the algorithm in Python 2.7 and used scikit-learn API (sklearn) to train our model. We use 880 composite figures and 1067 singleton figures randomly collected from our image corpus. We set $n = 10$ as the final parameter of feature descriptor. For each category, we randomly reserved 25% of the images for testing and applied the rest for tuning the SVM parameters by using sklearn's grid search method (part of the sklearn library). We shuffled the images to generate 10 different sets of training-testing pairs (10-fold validation). The testing set in each pair accounts for 25% data. After automatically scanning all possible combination of kernels and SVM parameters, the optimized model is run by the RBF kernel with gamma of 0.001 and penalty parameter of 10. Once the model parameters are well-tuned, all figure images were used to train the final model. Finally, we obtain 91.8% accuracy by 10-fold cross-validation on the entire training set comprising 880 composite figures and 1067 single figures. The recall and precision for each class are shown in Table 4.5.

Table 4.5: Evaluation of multi-chart figure classifier and first-generation figure-type classifier using 10-fold cross validation.

Figure Type	Precision	Recall
Multi-chart	92.9%	86.3%
Singleton	89.3%	94.6%
Equation	95.4%	95.1%
Diagram	84.2%	84.1%
Photo	94.5%	97.3%
Plot	91.5%	90.2%
Table	95.1%	93.1%

4.3 Figure Type Classification

The classifiers in the first generation pipeline are feature-based — "Bag of Feature + SVM". In the second generation pipeline, we introduce deep neural networks to improve the accuracy. The figure labels in the VizioMetrics dataset are predicted using the first generation pipeline. We will update the old labels together with the injection of new corpus.

4.3.1 Feature-based Classifiers

Our first-generation figure-type classifier is based on feature description. In 2005, Li et al. proposed "Bag of Features", an approach that extracts small patches (as features) from images, use them to build a codebook and then re-encode the images using a histogram (as the bag, explain later) [29]. we adapt the technique developed by Coates et al. [21] and extended by Savva et al. [82]. First, we normalize an image to a 128×128 grayscale image with a constant aspect ratio. Then, we randomly extract a set of 6×6 patches from each training image and normalize the contrast of each patch. Adjacent pixel values, and therefore adjacent patches, can be highly correlated. To increase contrast and better distinguish different patches, PCA whitening is applied on the entire patch set.

Next, we cluster the set of patches using k-means (k = 200) and to identify 200 common patch types, one for each cluster. A representative patch for each patch type, called a *codebook patch*, is derived from each cluster. For each training image, we generate a new set of patches by sliding a 6×6 window in one-pixel increments across the image. For each such generated window patch, we find the most similar codebook patch via Euclidean distance and increment a counter for that codebook. The set of codebook counters forms a histogram, and this histogram forms the feature vector used to train the classifier. To account for the global structure of common visualizations (e.g., axes are typically found on the left and bottom of the image), each image is split into four quadrants and a separate 200-element histogram is computed for each quadrant. The final feature vector of 800 elements is obtained by concatenating the four 200-element histograms.

To account for global patterns in plots (e.g., axes are typically found on the left and bottom

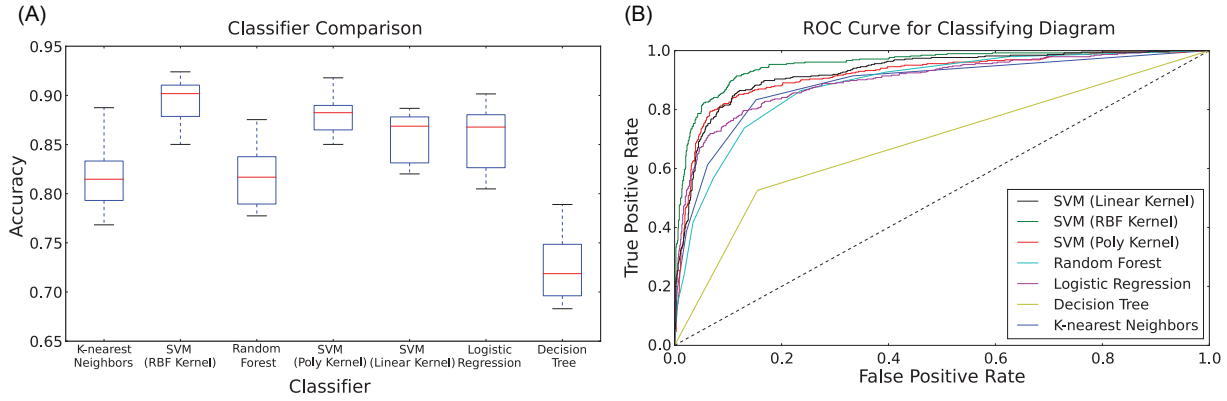

Figure 4.16: Comparison of classifiers: K-nearest neighbors, random forest, logistic regression, decision tree, and SVM with RBF, linear, and polynomial kernels, respectively. (A) The SVM with RBF kernel achieves the best performance evaluated by 10-fold cross validation. (B) The SVM with the RBF kernel also achieves the best performance compared to the linear kernel and polynomial kernel shown with the ROC curves.

of the image), each image is split into four quadrants and a 200-element codebook histogram is computed for each quadrant. The final feature vector of 800 elements is obtained by concatenating the four 200-element histograms.

4.3.2 Deep Neural Networks

Encouraged by the overwhelming performance of deep neural networks on classifying natural images, we investigate the utilization of deep learning for scientific figures. For the second generation figure-type classifier, we use two awarded architectures: AlexNet and Deep Residual Networks (ResNet) and train the networks using our own dataset. We extend our labelled figure types to eight categories: equations, photos, diagrams, tables, plots, electrophoresis gels, metabolic pathways and dendrogram. We train the networks using Caffe framework [47] installed on an Amazone EC2 instance (g2.2xlarge). We reserved 80% images for training, 10% images for validation in training phase, and 10% images for testing. Iteration stops when the accuracy of testing on validation data is steady and invariant to learning rate.

4.3.3 Experimental Result

We implemented Savva's approach in Python 2.7 using the same parameters as our first-generation figure type classifier. Instead of using WEKA, we trained our model using scikit-learn API (sklearn). We evaluated five different classifiers: K-nearest neighbors, random forest, logistic regression, decision tree, and SVM with RBF kernel. The corpus we used for training was randomly sampled from the PMC corpus. We manually labeled 3,271 images as one of five categories: photos (782), tables (436), equations (394), visualizations (890), and diagrams (769) and used these hand-labeled data to train the classifiers. We compared the accuracy of the five classifiers obtained by 10-fold cross validation and selected SVM with an RBF kerneal based on its superior performance performance (Figure 4.16(A)). To fine tune the SVM parameters (kernel, gamma, and penalty parameters), we randomly reserved 25% of the images for a testing set and trained the classifiers on the remaining 75% for each category, then used a grid search method to complete the task. Figure 4.16(B) shows the ROC curves for identifying diagrams. The optimized model is run by the RBF kernel with gamma of 0.001 and a penalty parameter of 1000. We focused on diagrams because the RBF kernel performs particularly well in recognizing diagrams. Once the model parameters are tuned, we evaluated the model by using the testing set and then trained the final model with all images. Table 4.5 shows the classification performance in each category obtained by 10-fold cross-validation. The final classification accuracy for all images is 91.5%.

Training a deep neural networks requires a large annotated dataset; thus we not only extend the categories but also expand the annotated images. We use our online labelling tool (see details in Section 5.2) to label more figures. The tool allows users to (1) correct the figure labels that are generated by the first generation classifier and (2) search figures with keywords and label them with the given keywords. We use the first mode to collect more equations, photos, diagrams, tables, and plots and the second mode to collect phylogenetic trees, metabolic pathways, and electrophoresis gels. We labelled 15,507 images from the tool together with 3,271 labelled images used to train the first generation classifier. Finally we collected 18,778 images: 1411 electrophoresis gels, 1871 equations, 1119 metabolic pathways, 3347 photos, 1308 phylogenetic trees, 2849 diagrams, 2193

tables and 4680 plots.

Table 4.6: Evaluation of figure-type classifier using hold-out testing set with 1878 images.

Figure Type	**Precision / Recall**		
	AlexNet	**ResNet-50**	**Bag Of Feature + SVM** [82]
Equation	98% / 97%	97% / 97%	97% / 97%
Photo	95% / 100%	95% / 96%	93% / 95%
Electrophoresis Gels	89% / 98%	96% / 97%	85% / 80%
Plot	98% / 95%	94% / 94%	90% / 91%
Table	100% / 97%	95% / 94%	95% / 94%
Diagram	88% / 92%	74% / 74%	75% / 74%
Metabolic Pathway	94% / 84%	84% / 76%	73% / 77%
Phylogenetic Tree	99% / 89%	87% / 93%	91% / 86%
Accuracy	**95%**	90%	88%

We tested two methods for training the networks: (1) train the networks from scratch, and (2) fine-tune the networks were pretrained on the 1.2 million images from ImageNet [54]. We received very similar results from the two methods so we just reported the one with better performance for each architecture (AlexNet trained from the scratch data and fine-tuned ResNet-50) in Table 4.5. In recent study, Siegel et. al fine-tuned the pretrained AlexNet [54] and ResNet-50 [39] with 60,000 figures annotated using Amazon mechanical turk [85]. They created seven categories and focused on classifying plot graphs such as bar plots, line plots, etc. They finally obtained 84% and 86% accuracies from AlexNet and ResNet-50 respectively. Different from [85], we obtained the highest accuracy from AlexNet trained from scratch (Table 4.6). We received a lower accuracy from the bag of feature approach when using the extended image set to train the model. The distinct decrease of recall and precision for diagrams is attributable to the increasing diversity of diagrams in the extended training set. In general, classifying diagrams and the associate sub-types are more

difficult than other types of figures due to the higher diversity, showing in all the tree models. Compared to the first generation classifier, AlexNet achieves an overwhelming performance on identifying diagrams.

4.4 Summary

In this chapter, we present a figure processing pipeline that automatically classifies figures into equations, diagrams, plots, photos, and tables. The pipeline includes a classifier to identify multi-chart figures, a dismantler to segment the multi-chart figures into singleton figures, and anther classifier to categorize the figures into different types. The feature-based multi-chart classifier achieves 91.8% accuracy and the dismantler successfully extracted 67.5% of the sub-images from 261 testing figures that contains two or more sub-figures. For our figure-type classification (8 categories), we obtain 88% accuracy from the first generation classifier based on bag of feature and SVM, and 95% accuracy from the second generation classifier based on AlexNet.

Chapter 5

VIZIOMETRICS ONLINE APPLICATION

Consider a biologist in search of the phylogenetic tree associated with a virus. Using a conventional academic search engine, she enters keywords (perhaps the name of the virus and the word phylogenetic), retrieves a list of candidate papers, and, inspecting the title for relevance, opens each paper for manual review. This process operates at the wrong level of abstraction, as the search is focused on a particular method that is associated with a visual encoding — a phylogenetic tree has a distinctive visual representation. Consider another case where a researcher wants to compare a number of different designs for solid-state laser diodes. She would like to find both scanning electron microscope (SEM) images as well as diagrams illustrating the designs, with goals of performing non-trivial analysis *across* figures: comparing the SEM photos with the corresponding diagrams (perhaps from a different paper), or a comparison of leakage currents by inspecting a set of plots showing the current-voltage curves. With both examples, keyword search followed by manual inspection of papers to gather specific visual results seems unnecessarily inefficient. We aim to use our classification pipeline to power a more efficient approach to this task using a figure-centric search application. We introduce VizioMetrics.org, a platform built on top of the PMC corpus. VizioMetrics.org includes functionality for three groups of users: (1) academic users performing search tasks, (2) researchers who specialize in computer vision for document understanding, and (3) scientometricians interested in understanding general communication patterns across the literature. For group (1), VizioMetrics.org provides a figure-centric search service that allows browsing and filtering by figure types. The figure labels are determined by our classification process, but the labels can be edited directly through the interface to improve accuracy over time. For group (2), VizioMetrics.org provides an efficient bulk-labeling interface to produce better training data for figure type classification and related viziometric analysis tasks. For group (3), VizioMetrics.org

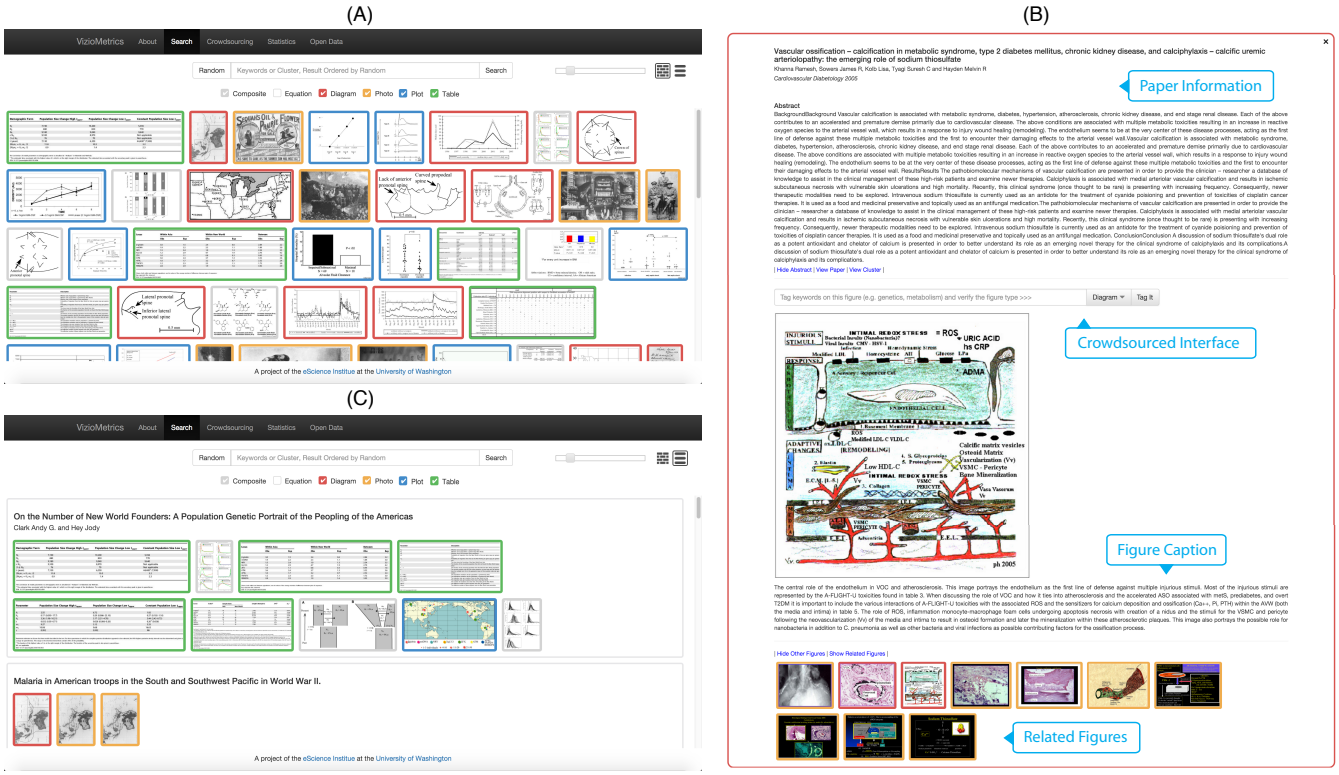

Figure 5.1: Screenshot of search engine interface (viziometrics.org). We use different colors to highlight different figure types (e.g., red indicates diagrams) (A) shows the grid layout, which is designed for reviewing many images (B) shows the alternative layout, which bundles figures from the same paper and related papers. Related papers are selected based on out and in-citations and then ranked by the ALEF score. This is made to look more like a paper, whereas the grid layout provides a general overview on a particular topic. (c) is the page showing figure and paper details. A simple crowdsourced labelling interface is embedded in the page to gather human labels.

provides a longitudinal analysis interface that allows aggregate analysis of the patterns of visual information in the literature. In this section, we first introduce the user interface of the search engine and the labelling platform and then describe the architecture of VizioMetrics.org.

5.1 Figure-centric Search Engine

Search tasks typically involve accessing a superset of relevant papers and manually scanning their contents for relevance. The text-based layout is not necessarily conducive for this ”scan for rel-

evance" task, and as a result the anecdotal algorithm is to "skim the figures." In all fields, key experimental results are presented in plots, complex scientific concepts are visualized by graphics with text, and photographs provide evidence and insight. Reviewing these figures can help users quickly understand the style of article or the methods used (e.g. statistical analysis, experimental demonstration, survey). In many fields like Cell Biology, a figure can be worth a thousand words that summarizes the entirety of the paper [56]. Using a particular style of schematics, plots, or photos can be indicative of a particular type of paper. For example, a phylogenetic tree indicates a phylogenetic analysis has been performed. The visual information can carry details that are insufficiently described in the text. Our hypothesis is that reviewing these figures as first class artifacts during search can help users rapidly identify relevant articles, draw associations between related articles, and focus attention on key results rather than overarching topics.

Figure 5.1 shows the user interface for the visual search application. At a basic level, users search for figures associated with a given keyword query. Each figure can be clicked to see metadata about the paper that contains it. This "inversion" of the search to emphasize figures before papers represents an important shift, even on its own, one that is shared by other recent tools, such as DiagramFlyer. In particular, the figures are more closely related to specific results, and are therefore, we hypothesize, more closely related to the intent of the user's search.

The search performs a free-text match on the caption text extracted from the full-text documents (and eventually, like DiagramFlyer, the extracted text from the image itself) and returns relevant figures via a free-text index on the underlying database that incorporates stemming and tolerance for spelling errors. The returned figures are ordered by the ALEF score as an estimate of impact. To facilitate browsing and prevent certain figures from dominating the experience, the user can shift to random order by clicking a button near the search box.

The returned figures are arranged in a grid layout (Figure 5.1(a)) to make better use of screen real estate and account for the widely varying shapes and sizes of figures. We color the figure border to indicate the type of the figure identified by the classifier; the legend for these colors appears at the top of the screen. Users can retrieve additional figures by scrolling down to the bottom of the page.

The most important feature in our approach is that users can restrict the search to figures of specific types by using the checkboxes just under the search box: photographs, tables, visualizations, diagrams, equation or composite. This faceted search feature is simple. We posit that this categorization of figures based on their semantics (as opposed to just their embedded or surrounding text) fundamentally changes the way users can interact with the literature. For instance, a clinical software engineer in a cancer lab may search for papers describing a cloud architecture for electronic health records that they can use to inform the design of their own system. Searching for "cloud" and "EHR" may return figures from relevant papers, but the precision will be low since the task is to find *specific architectures*. With this interface, restricting figure types to diagrams can improve precision. An alternative layout is bundling the figures from the same paper together and listing the papers (Figure 5.1(b)). This mode is designed for users who are looking for particular papers, but who may recall a memorable figure from the paper if not the title or author. Viewing article titles together with figures may help them narrow the scope. In addition, the figure marked by "star" at the top right corner is the recommended key figures of a paper. It may help the users catch the key graphical information of a paper. For figures with dense information such as composite figures, users can shift the slider to zoom in for close inspection or click the figure to review the figure caption, the source paper details, and other figures from related papers associated by the citation network. We bring the crowdsourced labelling function in this page showing the figure and paper details. We simplified the labelling interface (Figure 5.1(c)) to a quick click if the machine-label type is correct. For an incorrect case, it needs an extra move of modifying the figure-type in the drop-down menu. In addition, users are allowed to tag keywords of the figure with free text, assisted by autocomplete for reducing duplicates and typos.

5.1.1 Evaluation of Figure Search

We evaluate the relevance of the figure search for a *figure-based method search task*. This task consists of using keyword search for a particular method, with the intent of finding figure that represent the result of using that method. Anecdotally, we find this task to be both common in practice and poorly supported by paper-oriented search engines.

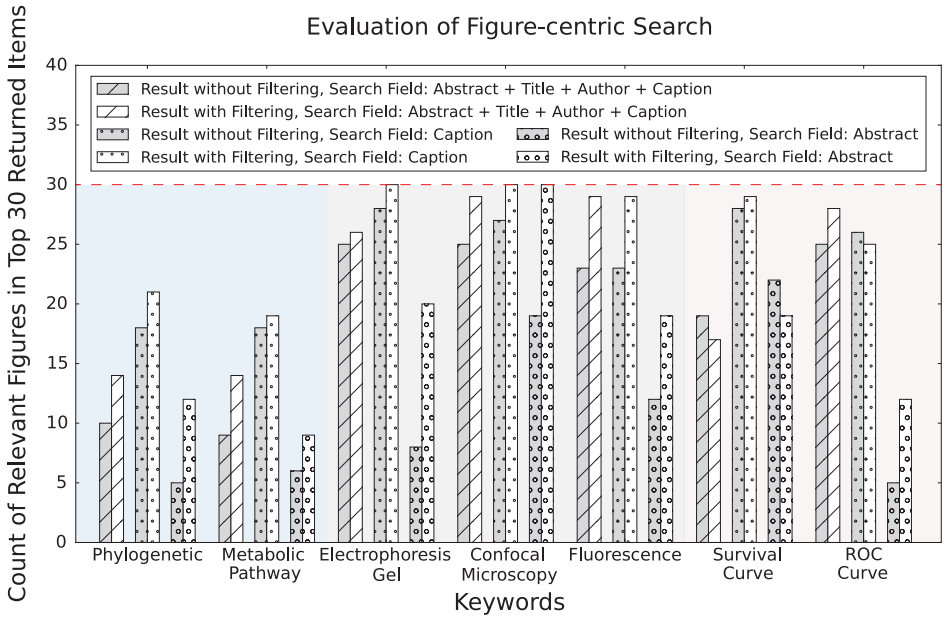

Figure 5.2: We selected 7 key phrases used to describe specific methods in biology that are associated with specific visual signatures. We report the proportion, of the top 30 returned figures, that correspond to the search term. When one searches ROC curve, the results should include ROC curves. We find that filtering improves the results in all cases except a few of the plot searches. We also find that, when restricting the search index to only captions, the results tend to be slightly better. The reason is that if a search term is mentioned in the abstract or title, then all figures in the paper are returned as results, lowering accuracy.

To evaluate the ability of viziometrics to support this task, we measure the proportion of top-ranked results that match the search term, using expert labeling as ground truth. For example, a phylogenetic analysis typically produces a particular type of tree that is recognizable to researchers. We report the proportion of the top 30 returned figures that correspond to the method in question. We choose the top 30 because it is the approximate number of figures shown in a page without the need to scroll.

We consider the following questions: 1) Does the search interface tend to retrieve relevant figures for figure-oriented search tasks? 2) Which fields should be indexed to maximize accuracy? 3) Does filtering the results for an expected figure type (using the results of our classifier) improve accuracy?

To answer these questions, we use seven key phrases associated with specific figure types as our search terms: `phylogenetic`, `metabolic pathway`, `electrophoresis gel`, `confocal microscopy`, `fluorescence`, `survival curve`, and `ROC curve`. For each term, we evaluate different indexing strategies: caption only, abstract only, or abstract, title, author, and caption. Finally, we consider what effect filtering by figure type has on accuracy. For example, when searching for `phylogenetic`, the figures associated with the term are typically diagrams, so ignoring all other figure types except diagrams should improve accuracy. Other search terms are similarly associated with a dominant figure type: `phylogenetic` and `metabolic pathway` are associated with diagrams, `electrophoresis gel`, `confocal microscopy`, and `fluorescence` are associated with photos, and `survival curve` and `ROC curve` are associated with plots.

Figure 5.2 shows the results. Overall, 50% to 100% of the results are relevant for each search term under the best conditions. We find that caption-only indexing provides the highest accuracy. The reason is that if a search term is mentioned in the abstract or title, then all figures in the paper are returned as results, lowering accuracy. We find that properly filtering by figure type further improves the accuracy, typically including 2-10 additional relevant figures in the top 30 results. However, in some cases filtering reduces accuracy; in these cases the classifier's imperfect type assignment is the culprit. Despite the improved accuracy achieved by caption-only indexing, we index all fields in the current application to ensure that we return relevant papers.

The search engine is available online at `www.VizioMetrics.org`. Anecdotally, we have had users report that they use the interface to find figures for textbooks and presentations. They describe the system as the "google images" for scientific figures.

The one significant limitation of Viziometrics.org is the available content. Most scientific papers are held behind publisher paywalls. In our first version of the system, we have included figures from PubMed Central. Although this open corpus includes millions of figures, it only represents a small proportion of medically related research. Our hope is to extend the corpus to all disciplines, but this goal will depend on improved access to the scholarly literature.

5.2 Crowdsourced Labeler

The viziometrics analysis tasks we aim to support rely strongly on the availability of human-labeled data. We hand-labeled thousands of images when training our dismantling algorithm for separating composite images and our classifiers. Going forward, additional labels for figure sub-types, content tags, and information extraction techniques will require even more human-labeled data for training. For example, a line chart in oceanography may be a "depth chart," while a line chart in machine learning could be an ROC curve. The level of expertise needed now and in the future motivated an efficient, intuitive, and expert-oriented interface for bulk-labeling of figures.

Due to the expertise required, these labelling tasks are not directly appropriate for Mechanical Turk. We therefore include the labelling interface as part of the core system in the interest of enticing our (expert) users to contribute to the labelling task directly. Figure 5.3 shows the interface for (A) 6-cat labeler and (B) free-text labeler. Labelers are shown an instructional page as a first step, and can access the instructions at any time by clicking the "Need Help?" button.

The 6-cat labeler asks a user to select figures of a certain type (e.g. Diagram) chosen randomly in each round. Figure can be included or excluded in the selection by clicking. We hypothesize that an approach of "one label, many images" will allow the user to quickly become efficient at recognizing a specific type rather than spending too much time considering a specific figure. The decision to make on each figure is binary, and "difficult" figures can simply be ignored. We suspect that this approach will produce high accuracy labels, with a possible downside that difficult-to-classify images will be consistently ignored, perhaps complicating training tasks.

Clicking the "Next" button will submit the choice and go to the next round. The figures that were not chosen will stay in the new round. We expect the unchosen figures will eventually meet their categories if the user stays for enough rounds. We feed 20 figures in each round to help prevent unconscious mistakes caused by repetitive tasks. Since we group the candidate figures according to their machine-labels, the user's task is often confirmatory. In this case, using the "Label All" button can save time. Although this approach may lead to confirmation bias in the human labels, but we find anecdotally that mistakes in the machine labels stand out and attract attention. When

Figure 5.3: Screenshot of the bulk-labelling interface: (A) 6-cat labeler and (B) free-text labeler. (a) Instructions for using the interface. (b) Indicator for the type of figure the user is asked to label (e.g., photos). (c) The total number figures that have been labelled by the user. (d) Controls to allow users to label all figures directly. (e) Controls to allow users to refresh the pool of figures. (f) Submits the result. (g) Zoom control to afford fluid inspection of figures and groups of figures. (h) Input text box (i) Submit the result of selected figures. When the user modifies the keywords, it changes to a search button.

the user is satisfied that each figure in the set is labeled correctly or ignored, clicking the "Fresh" button can retrieve a new pool of unlabeled images. We don't set the end of round so the user can stop in any round.

The free-text labeler allows a user to find specific figures, e.g. survival curve, dendrogram, etc. and quickly label them. The system searches for figures associated with a given keyword query. From the returned images, the user can select the figures that are truly relevant to the given keyword. Clicking the "Submit'' button will send the choice to the database and the selected figures will be labelled with the search keyword. The keywords obtained from the free-text labeler will be used to create new categories for training new classifiers or developing other viziometrics tool. In this study, we utilize the crowdsourced labeler to build a larger image set for training deep neural networks (Section 4.3.2) and gather dendrograms for developing phylogenetic tree parser (Section 6).

5.3 Backend Architecture

VizioMetrics.org's architecture proceeds in two sub-systems. The offline data backend extracts visual information from a corpus and stores them in a database used to power the online system. The online system includes three applications for different audiences (researchers searching the literature using figures as the central facet and researchers mining figures from the literature). Figure 5.4 shows the system architecture. Our design goal is to offer a reusable platform to support viziometrics applications.

5.3.1 Data Backend

The VizioMetrics database consists of paper metadata, bibliometrics data and figure data. The paper matadata are extracted from full text documents, and the bibliometrics data are provided by our collaborator Eigenfactor.org (Section 3). The figure images are stored in the Amazon S3 server. The figures are fed into a figure processing pipeline (Section 4) for classification. We run figure processing in parallel. The final results are pushed into the VizioMetrics database (currently

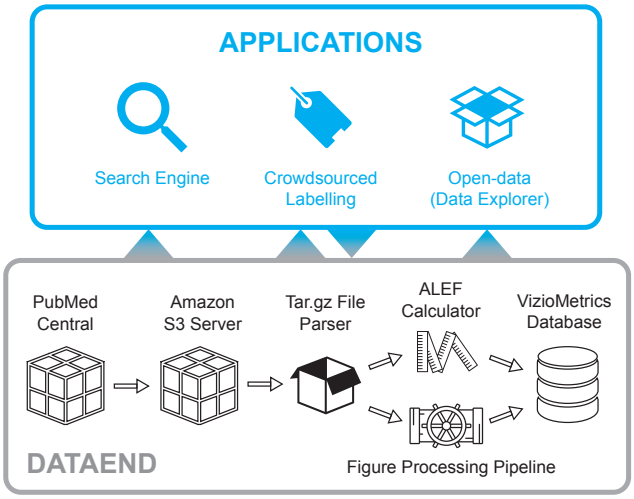

Figure 5.4: The Architecture of VizioMetrics.org. The gray box illustrates the processing system on which VizioMetrix is based. This includes a data pipeline, which parses articles files, classifies figures, and calculates the article influence. The data pipeline then pushes the results into the VizioMetrix database. The blue box lists the three applications powered by the database.

implemented in MySQL) together with figure information such as the figure id, the file path on S3, the image size, etc as the figure data. Figure 5.5 shows the overview of the VizioMetrics Data. We join the paper data with the figure table, and materialize the result for performance reasons. Due to the limitation of full-text search in MySQL, we integrate Solr with MySQL as a free-text indexing solution to achieve reliable performance.

5.3.2 Search Engine Backend

The Search Engine backend is responsible for data ingestion. Whenever a client sends a GET request with search keywords, the backend system will query the database to find the literature containing the keywords in title, abstract and figure captions. The system indexes the authors, titles, abstracts and figure captions of the corpus of papers; keyword searches probe this index to find relevant images. Result figures are ordered by their ALEF scores, helping to reduce attention on low-impact papers and returned in in JSON format. Since the figure files are stored in Amazon

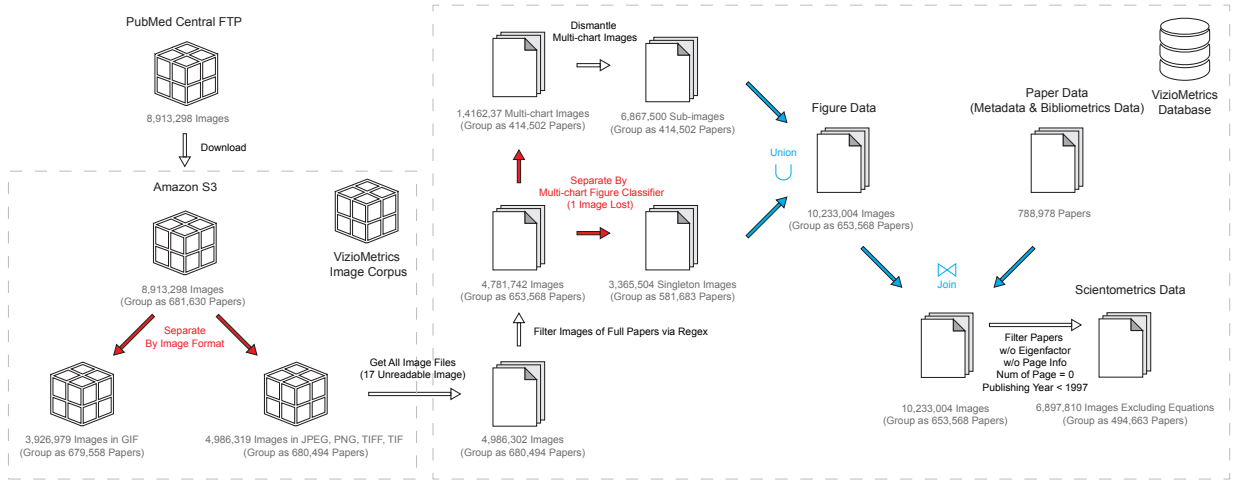

Figure 5.5: VizioMetrics Data Overview. We extracted all papers from PubMed Central (PMC) repository, an archive of biomedical and life science literature. It offers free access to approximately 1 million articles. Every article is packaged with its PDF files, full text documents, and figure images. We extracted the figure images and dumped them in Amazon S3 server. Every image has a unique key for access via AWS SDK. This diagram shows the exact numbers of images involved in the filtering steps (Section 3) and lost in the figure processing pipeline. We parsed the full text documents to get paper titles, abstracts, citations, etc together with the bibliometrics data acquired from Eigenfactor.org and stored this information in our database. We joined the figure data and paper data as the main table for the use in our search engine. We also compile a subset of data with more filtering steps for scientometric research.

AWS S3, we store URLs only and use S3 to deliver the images to the client. The front end fetches the returned JSON and renders the layout in the user's desired mode (a grid or a conventional list). We return only the first 100 figures ordered by paper impact in descending order to achieve reliable performance. When the user scrolls to the end of the set of figures, a new GET request will be issued to retrieve the next 100 figures. When a user clicks a figure to get further information, another GET request is sent to retrieve relevant figures from associate papers. The recommendation of relevant figures is based on the hierarchical clusters of articles provided in the Eigenfactor data. The system returns the figures from the papers that are in the same cluster of the paper containing the selected figure and orders them by their ALEF scores. See Section 3.4 for the details of Eigenfactor data and the recommendation techniques from [103].

5.3.3 Crowdsourced labeler Backend

We provide two labelling modes: (1) 6-Cat labeler that labels figures as one of the five categories: photo, table, equation, visualization, diagram, and composite figure and (2) Free-text labeler that tags free-text keywords on figures. The crowdsourced labeler backend responds to GET and POST requests. Whenever a GET request is received, it returns 20 figure image URLs. For the 6-cat labeler, the 20 figures are in the same category randomly selected by the system; for the Free-text labeler, the system uses the search engine to get 20 random figures that are related to the given keywords. When the user returns the labelling result, the system will push the user's IP address, the figure id, and the given label into the database. The ground-truth type of a figure will be determined via voting. Since we have a very large image inventory and is growing continuously, one figure is likely to be labelled once and never be selected as a candidate again, which affects the credibility of the ground-truth data. Hence, we increase the probability of selecting images that have been labelled previously. The crowdsourced labelling data are used offline to improve our machine learning models and also open to the public for academic use via our REST APIs.

5.4 Summary

In this chapter, we present VizioMetrics.org, a platform for mining millions of figures from the biomedical sciences. Our hope is that the platform will catalyze future research for improving scholarly search and facilitating large-scale analysis of these figures and new figure-centric applications. VizioMetrics.org provides a figure-oriented search service for general academic users and an open data resource for researches interested in mining scholarly figures. We also develop a crowdsourced labelling application for further labeling and subsequent improvement of our machine learning methods and methods of others. This platform is needed since mechanical turkers do not usually have the domain knowledge for labeling professional figures. We hope this platform reduces the activation energy needed to analyze scholarly figures at scale and provokes new and exciting questions.

Chapter 6

PHYLOPARSER: A HYBRID ALGORITHM FOR EXTRACTING PHYLOGENIES FROM DENDROGRAMS

Scientific results in the biomedical literature are frequently presented visually with figures, diagrams, and tables, but the information contained in these objects are inaccessible to text-oriented computational approaches. As part of the viziometrics project, we are working to develop a general framework for information extraction from the visual literature using computer vision and machine learning approaches.

In this section, we focus on extracting information from phylogenetic trees. These trees are used extensively within genetics, cladistics, conservation biology, medicine, public health and many other areas of biology [106] to organize evolutionary relationships between species into a hierarchy rendered as a dendrogram. They are used to track the evolution and spread of viral infections [35], migration of species [41], and for comparing genetic sequences [52].

Public repositories for phylogenetic information have been created including TreeBASE [66] and MorphoBank [70]. These databases are intended to organize and aggregate results to help build scientific consensus ahout the tree of life [12]. However, these databases are relatively new, and are not comprehensive for at least two reasons: Results from older papers are missing entirely, and even among current papers, there is no mandate to use these repositories. In 2017, there are more than 40 thousands phylogenetic trees available on PMC, which is three times the number of trees available on TreeBASE. More broadly, policies designed to encourage researchers to clean and share their data have had limited success [105]. We aim to use PhyloParser to construct a new database of phylogenetic information, with goals similar to that of TreeBASE, but to derive it automatically from the scientific literature itself to increase coverage and reduce human effort.

Previous approaches to this problem either rely on human input or are sensitive to noise, mak-

ing them inadequate for our purposes. For example, the interactive approach proposed by Laubach et al. [57] would require hundreds of hours to process typical datasets. Previous automated approaches rely on line-tracing techniques that are sensitive to noise. The figures in the literature are extremely heterogeneous: they may involve complex annotations and background formatting, vary in size and resolution, and use inconsistent spacing between lines, text, and other elements. These complications prevent any one method from being successful in all cases. Our approach, in contrast, is to combine machine vision approaches with a topological "grammar" of how these trees are constructed in order to significantly reduce errors.

We automatically recognize the fundamental components of dendrograms: horizontal branches, vertical branches and the text of species names. We then use Hough transforms and convolutions to extract these components with high recall, then filter false positives by applying topological heuristics about how these components fit together. The tree structure can be recovered by assembling these components both both top-down and bottom-up. This approach allows us to ignore noise at the pixel level and enables partial recovery of a dendrogram that may have poor global quality but good local quality; we find this trait to be critical in extracting information from characteristically noisy images in the literature. For instance, Hughes et al. used a dataset selected to favor their approach [45]; even on this curated dataset, our approach extracts 11% more perfectly recovered instances, approximately 80% information from the imperfect cases, and runs significantly faster.

The errors made by our algorithm tend to occur when the resolution and the quality of the images are low, when the text and the lines overlap, and when tree branches are of a color very close to the background color. None of these sources of error appear insurmountable.

We make the following contributions.

- We present a classification approach to recognize phylogenetic tree diagrams in the literature using deep neural networks.

- We present a hybrid algorithm for parsing dendograms that recognizes the basic elements of the diagram and then assembles them topologically to be more resistant to local noise.

- We evaluate these methods on datasets from prior work as well as a new dataset extracted from one million papers in PubMed.

- We organize these methods into an end-to-end system called PhyloParser that produces new relationships in a machine-readable format.

- PhyloParser and all datasets we use in our experiments are available online to enable future research.

6.1 Proposed Methods

Our tree parser is designed for trees with the following prerequisites, which appear to be met in practice:

- The tree is constructed by horizontal and vertical lines.

- The root is on the left and the leaves are on the right.

- The background color is lighter than the tree structure and the text.

Figure 6.1 illustrates the pipeline of our parsing algorithm. We divide the algorithm into four stages: preprocessing (grey), tree component extraction (blue), text extraction (pink), and tree reconstruction (green). During preprocessing, we remove any color background if it exists, then separate the tree structure from the text. During tree component extraction, we recognize tree components including vertical lines, horizontal lines, corners, and joints. Then we reorganize the components based on their connectivity. During the text extraction, we locate text regions and convert them into text using standard optical character recognition (OCR) to acquire species names. In addition, we associate each species to the corresponding tree leaf. During tree reconstruction, we assemble all components together to recover the entire tree. In the following sections, we will describe each stage.

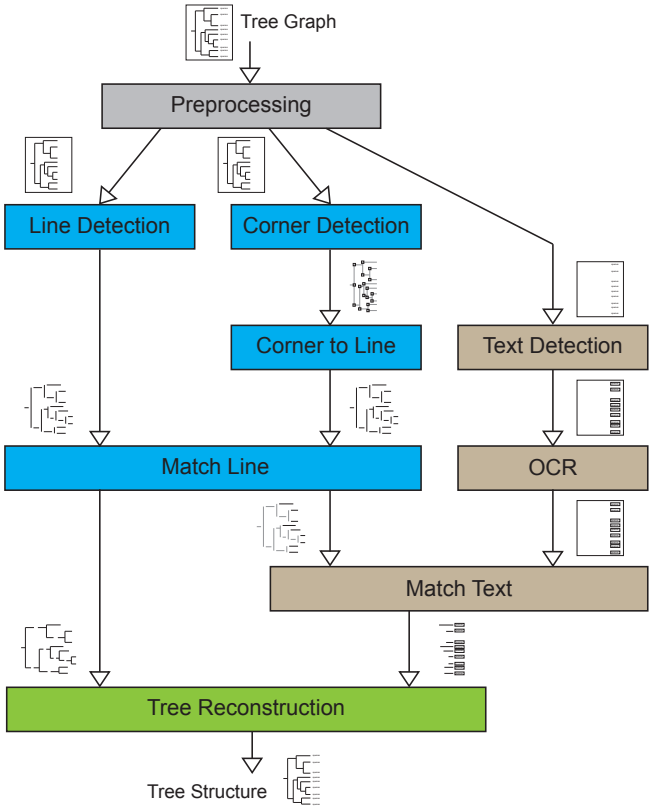

Figure 6.1: Pipeline of parsing a tree diagram. There are four stages: preprocssing, tree component extraction, text extraction, and tree reconstruction. First, we remove color background and separate text from tree diagram (grey box). Next, we extract tree components (blue boxes) and specie names (pink boxes) from the image. Finally, we connect the components to recover the tree structure (green box).

6.1.1 Parsing Phylogenetic Trees

Figure 4.1 illustrates the pipeline of our parsing algorithm. We divide the algorithm into four stages: preprocssing (grey), tree component extraction (blue), text extraction (pink), and tree reconstruction (green). In the preprocessing, we remove color background if it is existed and separate the tree structure and text. In the tree component extraction, we recognize tree components including vertical lines, horizontal lines, corners, and joints. Then we reorganize the lines based on their connectivity. In the text extraction, we locate text regions and convert them into text using standard optical character recognition (OCR) to acquire specie names. In addition, we associate the species

to the corresponding tree leaves. In the tree reconstruction, we assemble all components together to recover the entire tree. In the following sections, we will describe each stage.

Preprocessing

Our parsing algorithm is based on identifying horizontal and vertical line segments locally, then topologically connecting these segments into a global tree. We find that high-accuracy line detection at this stage is critical for achieving a low-error result.

Images is first converted into gray scale. To make our algorithm scale invariant, we first resize the input image to the dimension in which the line thickness is normalized to be lower than four pixels. Without this step, thick lines can be mistaken for dark patches that will be removed in future steps. The line thickness is determined by iteratively applying a morphological opening operation: an erosion to reduce thickness followed by a dilation to expand thickness, which can "open" connections between elements. For each iteration, we increase the morphological kernel size. The iteration terminates when the sum of pixel values in the image is half of the sum of pixel values acquired from the original image (indicating the tree structure has been completely erased), or stops after 15 iterations. Too many iterations can make it difficult to distinguish the lines from a dark background. In this case, we do not resize the image.

A common source of errors for line detection is text, so after the opening process, we need to separate text from the tree structure. We acquire contours by using the simple contour finder in OpenCV, then use the heuristic that the tree structure is typically the largest contour in an image. This largest such contour is used to build a *tree mask*, and the remaining contours are used to build a non-tree mask. We apply this tree mask to distinguish the main structure of the tree from other sources of lines, including text.

Next, we remove color patches in the background. These patches decrease the gradient of tree boundaries, making it difficult for the line detector to identify segments reliably. To detect color patches, we use the Canny edge detection to highlight all boundaries, followed by a morphological close operation with a 5×5 rectangular kernel to backfill the tree structure. We locate the pixels within these boundaries to identify color patches. Next we extract the colors from these background

pixels and whiten all pixels with the same colors in the original image.

Finally, we use a bilateral filter to sharpen edges, which improves line detection specifically for thin lines.

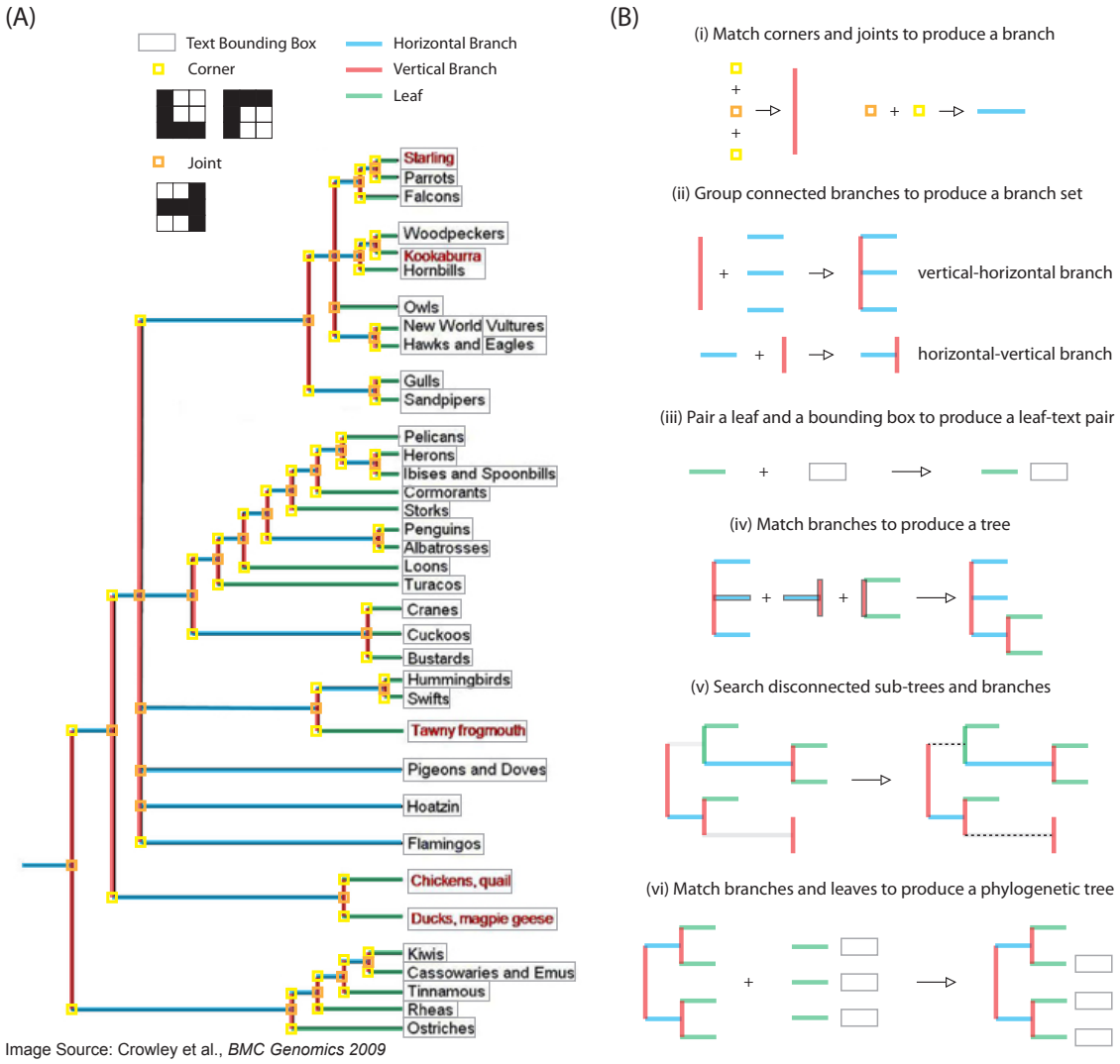

Figure 6.2: (A) Tree components and (B) reconstruction logic. Our parsing algorithm is based on accurate line detection. Beside using classical Hough transform to detect lines, we also extract lines by detecting corners and joints as shown in part (i) of (B). Texts are located by contour finder and converted by Google Tesseract. In (B) We visualize the concept of "Match Line" in part (ii), "Match Text" in part (iii) and "Tree Reconstruction" in part (iv) to (vi). (The original figure created by Nathalie Boone et al. [22], used under CC BY 4.0/ Modified from original.)

Tree Component Extraction

The main task in this stage is to extract vertical lines and horizontal lines. We first use the non-tree mask to whiten irrelevant pixels in the image and then detect lines using two approaches: Hough transform and endpoint detection.

The Hough transform can detect lines at given angles. We only need vertical lines and horizontal lines. The text pixels that are not successfully excluded by the mask can produce many short lines. We identify and drop each short vertical line l_{ver} and each short horizontal line l_{hor} according to the following two heuristics:

`drop` l_{ver} `if` $length(l_{ver}) < 8 + height(image)/100$

`drop` l_{hor} `if` $length(l_{hor}) < 3 + width(image)/100$

We determined the threshold values 3 and 8 experimentally. We set a higher threshold for vertical lines because the minimum length of true vertical lines is usually longer than the true horizontal lines in the tree topology. We determined that our experimental results are not sensitive to this parameter in the range 6 to 15 pixels. This filtering step does remove a portion of true lines and thus prune a tree. In the later stage, we will recover the missing part via bottom-up reconstruction methods.

A Hough-transform-based line detection, such as the one we use here, does not guarantee 100% recall. To increase the recall, we capture the corners and joints (see Figure 6.2(A)) and pair them to acquire vertical lines and horizontal lines. We first binarize the image with a threshold of 180. We use simple convolution with the three masks shown in Figure 6.2(A) to expose top corners, bottom corners, and joints, refined by two thresholds: (1) a high-pass threshold to include highly responsive pixels and (2) a low-pass threshold to eliminate highly responsive pixels that are entirely surrounded by black pixels. Next, we pair top corners with bottom corners to create vertical lines as well as corners with joints to create horizontal lines (Figure 6.2(B)(i)).

Since multiple lines can be detected from a single thick line, we group those lines that come from the same source to avoid duplicate pairs in the next step. Figure 6.2(A) shows the ideal result of line detection and corner detection. Next, we connect vertical lines and horizontal lines to

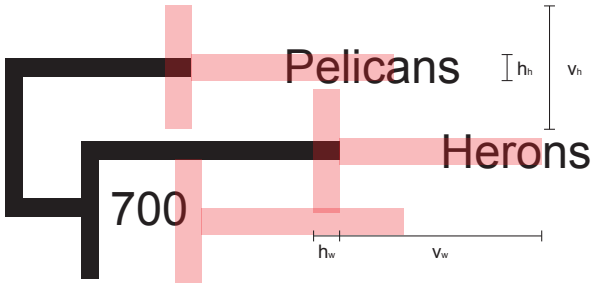

Figure 6.3: To identify leaves, we train a model based on spatial features between lines and text. We design the features to capture the unique pattern of leaves: a vertex followed by text. We extract the raw pixels (colored red), along with the horizontal distance between the right endpoint of a leaf and the mean x-coordinate of the endpoints of all leaves. To determine these features, we set $h_w = 3$, $h_h = 3$, $v_w = 15$, and $v_h = 9$.

create branch sets (Figure 6.2(B)(ii)). For each vertical line, we associate it with right connected horizontal lines as a vertical-horizontal branch (v-h-branch). We associate each horizontal line with the right-connected vertical line as a horizontal-vertical branch (h-v-branch). These branch sets will be used in the tree reconstruction process. The horizontal lines that are included in v-h-branches but have no right-connected vertical lines are defined as "leaves," which will be used to connect species names in the text extraction stage. Since the leaves can generated from the remaining text, we further use a binary classifier to verify the leaves. The features selected are based on the assumption that a leaf typically has an endpoint on the right followed by text. As an example of how these features manifest in practice, consider Figure 6.3. We use the pixel values in the red area as the image feature, together with the horizontal distance between the right endpoint of a leaf and the mean position of the right endpoints of all leaves. We compile a training set containing 3368 positive examples and 1201 negative examples, extracted from 100 tree diagrams (not included in the test set). We tested several classifiers and finally chose a random forest that gives the highest accuracy of 93.4%. Most of the false leaves are produced from text that are extremely close to the vertical lines. The classifier is effective to identify these "fake" leaves.

Text Extraction

We use the tree mask to remove the tree diagram as the first step of text detection. Second, we again use the contour finder to segment characters, strings, and any other irrelevant items such as arrows, numbers, etc. For each contour, we generate a bonding box for further use. Here, we eliminate tall and thin bounding boxes that are unlikely to contain text. We define these boxes as those satisfying two conditions: (1) the aspect ratio (height / width) is greater than 10 and (2) the height is also greater than 10.

Third, we generate leaf-text pairs (Figure 6.2(B)(iii)). For each leaf, we associate it with the bounding boxes locating right to the leaf. For the case that a bonding box links to multiple leaves, we divide the bounding box in the following methods: (1) Divide the bounding box based on nearby bounding boxes that suggests a reasonable division (see lower leaves in Figure 6.4). In this example, we separate "New World" from "Hawks and" based on the known patterns "Vultures" and "Eagles". (2) For the case without such hint (see upper leaves in Figure 6.4), we divide the bounding box at $(y^{i}_{leaf} + y^{(i+1)}_{leaf})/2$, where i denotes the associated leaf and y denotes the corresponding location on y-axis. We found that approximately 50% of our test images need such process to separate at least one bounding box.

Text in bounding boxes are converted using standard OCR methods [88]. For the leave not associated with any bounding box, we scan the area to the right with 10-pixel-high box to seek missing text. The text successfully associated with leaves will be used as the species names in the next stage, while the text not pairing with any leaf (an *orphan* text box) will be used to identify missing leaves.

Reconstruction

We reconstruct the tree by connecting h-v-branches and v-h-branches with their common lines. However, this step does not guarantee a perfect reconstruction because some vertical lines and horizontal lines are likely missing in the step of detection. In this case, we obtain several sub-trees. Each of these sub-trees will either be missing a horizontal line at its root, or will have at least one

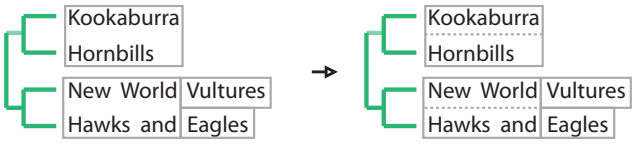

Figure 6.4: Method of separating cross-line bonding box. Text in different lines can be bonded together if they are very close. A reasonable segmentation can be deducted from (1) other bonding boxes or (2) corresponding leaves. We seek the first solution prior to the second solution, because the perfect segmentation from the second solution only applicable when all corresponding leaves are found.

broken branch. A broken branch is a vertical line not connected to at least two leaves on its right. To overcome this issue, we develop two methods to search for the missing connection:

A. Search for any existed sub-trees or vertical branch in the right area of the broken branch.

B. Search for any orphan text boxes in the right area of the broken branch.

Figure 6.2(B)(v) shows an example in which three sub-trees are not linkable because of undetected lines (highlighted in light grey). We reconnect the upper sub-tree using method A and the lower vertical branch using method using method B. Method C handles a particular tree style that a tree does not use horizontal lines to tip species, for instance the lower part in Figure 6.2(B)(v). For a broken branch, we associate it with the orphan text boxes within a distance of 15 pixels, defined by observation. Method C does not handle a rare case that horizontal lines are used for both top and bottom leaves but not for the middle leaves in a multifurcating tree.

The final step of tree reconstruction is merging the recovered tree structure and species names (embedded in leaf-text pairs and orphan text boxes). We have associated the text with the leaves or the vertical branches in the previous steps, so we only need to traverse the tree structure to produce the final tree string in the Newick format.

6.2 Evaluation

To evaluate our tree parser, we create a test image set randomly collected from the 1308 phylogenetic trees (described in Section 4.3.3). The test image set includes only trees constructed by horizontal lines and vertical lines, i.e. circular trees and cladograms are not included. In addition, we do not include low-resolution trees with any ambiguous branches and species names that are not readable by human. Despite that our approach is able to recover these trees partially, we are not able to create an absolutely correct ground-truth for a fair comparison. Finally, we compiled a test image set consisting of 141 phylogenetic tree images.

We first compare our result to [45] as the baseline. To pursue a fair comparison, we manually divided the 18 multi-tree images and preserved the tree diagram with provided ground-truth data. Compared to the 37 samples of successful recognition reported by [45], our approach perfectly extracts 48 tree structures from the TreeRipper dataset in which 33 perfect samples are from the 79 singleton tree images (Table 6.1).

PhyloParser is not designed to compete with perfect extraction. Our approach is developed to acquire phylogenies from big scholarly figures with high error-tolerance. We intend to correct remaining errors through an aggregate statistical analysis over a large corpus of trees as part of future work. To evaluate our approach more thoroughly, we use the Zhang-Shasha distance (ZSD distance) [108] to measure the tree edit distance between the recovered tree structure and the ground-truth. ZSD distance considers three operations:

- Change one node label to another
- Delete a node. All children of the deleted node become children of its parent
- Insert a node

Different order of siblings is identified differently when measuring ZSD. It does not agree with the essential property of a phylogenetic tree, i.e. changing the order of siblings does not change the meaning of phylogeny. Nevertheless in our study, we can ignore this discrepancy because

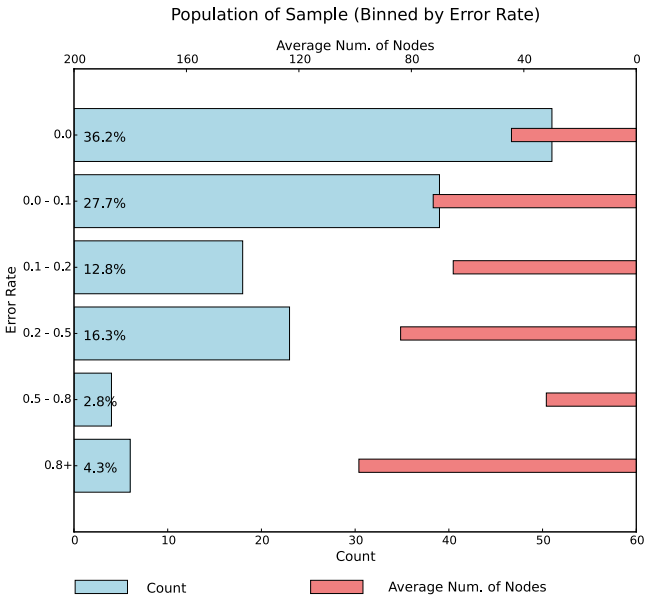

Figure 6.5: Histogram of reconstructed trees categorized by error rates. The wide bars show the figure counts and the thin bars show the average number of nodes. We obtain 51 perfectly reconstructed trees and 108 reconstructed trees with error rates below 0.2. The performance of our algorithm degrades when the size of tree increases.

our algorithm does not switch siblings during reconstruction. We normalize the ZSD to the total number of true nodes for each phylogenetic tree, defined as an error rate. The error rate of zero indicates a perfect reconstruction. The normalized value can be larger than one when the algorithm produces many false positive branches and leaves. We do not evaluate the performance of the Google Tesseract OCR engine, because it is not the main contribution of this study. Thus we name all species the same in our evaluation.

We evaluate our main tree reconstruction approach and the two methods for searching missing leaves, branches, and sub-trees. Table 6.1 shows the result obtained by using different combination of searching methods. *Base* denotes the line-based reconstruction algorithm; *A* is the method for connecting sub-trees and vertical branches; *B* is the method for associating orphan text boxes. We obtained the lowest mean error rate of 0.148 from *Base* + *A* + *B*. It can be interpreted as that approximately 15 nodes are missing, incorrectly located, or mistakenly created from a tree containing 100 nodes. Method B is a trade-off strategy because it can mistakenly create false

Table 6.1: Evaluation of the tree parser using TreeRipper dataset (100 images) and our own dataset (141 images). Hughes evaluated TreeRipper using 114 test images but provides a subset of 100 images for the use of branch mark.

Figure Type	**Error Rate**	**Count of Perfect Reconstruction**
PhyloParser Dataset		
Base	0.238	39 / 141
Base + A	0.156	**54** / 141
Base + A + B	**0.148**	51 / 141
TreeRipper Dataset [45]		
TreeRipper	N/A	37 / 114
Base + A	**0.116**	**48** / 100
Base + A + B	0.120	44 / 100

leaves: we receive a lower error rate on average but with fewer perfectly extracted trees compared to *Base + A*.

Figure 6.5 shows the binned result in which each bin represents a group of trees corresponding to a range of error rate. The wide bars denote the counting numbers of trees. We obtained 51 (36.2%) perfect recovered trees and 108 (76.7%) trees with error rate below 0.2. The thin bars denote the average numbers of nodes for the groups of trees.The average number of node is 45 for the perfect bin (error rate = 0), which can be seen as a balanced binary tree with approximately 20 leaves. Our algorithm performs better on small trees rather than large trees. The main reason for the degradation is that components in large tree diagrams are dense or even occluded. These large and dense trees are probably a consequence of page limits for journals. The other chunks of failures are trees with edges in extremely light colors. These lines are not detected by the Hough transform and neither are the corners and joints.

Figure 6.6 shows the qualitative results of three phylogenetic tree diagrams. We show the original figure on the left and the regenerate figure on the right for each sample. The left and the

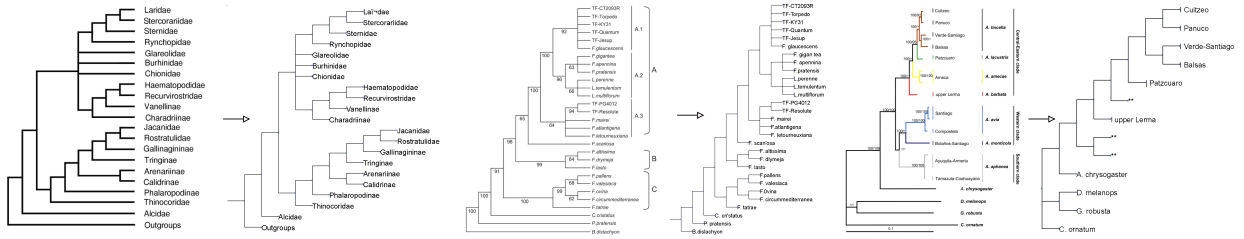

Figure 6.6: Qualitative results (Left: original figure, right: regenerate figure). The left and the middle phylogenetic tree diagrams are considered perfect in our experiment without evaluating OCR results. A few species names are not converted correctly by Google Tesseract. The right sample shows a failure example that the sub-trees highlighted by light colors are missing. In the regenerate figure, we use "**" to denote a broken branch. (Original figures created by Ericson et al. [27], Hand et al. [37], Pérez-Rodríguez et al. [73], used under CC BY 4.0/ Modified from original.)

middle samples are considered perfect in our evaluation; the right samples are partially recovered, because the light lines are not successfully detected. The Google Tesseract performs well for these tree diagrams. Only a few species names are not correctly converted.

6.3 Summary

In the chapter, we present PhyloParser, a framework that automatically identifies phylogenetic tree figures from scientific literature, extracts the key components of tree structure, and reconstructs them to recover the raw data of species relationships. We obtained an average error rate of 0.15 from our testing image set containing 141 tree diagrams collected from scientific literature. Our tree parser does not handle circular trees and cladograms, but we plan to extend our algorithm to broader styles of tree diagram. To improve the OCR results, we plan to extend the Tesseract dictionary by collecting specie names from NCBI taxonomy database. We are working to extract phylogenies from big scholar data and store them in a database and fix the error nodes by comparing overlapping trees. In future work, we aim to construct a database of species relationships automatically from the scientific literature, validate these relationships against manually constructed databases, and answer questions about the coverage and veracity of the results, and how the confidence of scientific results has changed over time. We also plan to build a scholarly search engine

particular for phylogenetic research.

Chapter 7

EXPLORING VISUAL PATTERNS FROM PUBMED CENTRAL

In this section, we present an initial exploration of viziometrics by analyzing the PMC corpus. We use the classified figures to study patterns in the use of visual information across scientific domains, across publication venues, and over time. We also used the classifications to examine the effect on scholarly impact. More broadly, we are interested in better understanding how complex results are communicated across disciplinary boundaries and to the general public, and how this communication channel can be optimized to increase the bandwidth of scientific discourse. Our method of longitudinal analysis of all figures in a domain is generalizable both to other domains and to other questions related to demography, editorial trends, narrative style, and influence. We provide preliminary results using this method and discuss the findings in order to provide a foundation for the two questions above.

A key result is a link between the use of scientific diagrams (schematics, illustrations) and the impact of the paper, suggesting that high-impact ideas tend to be conveyed visually. We conjecture two possible explanations for this link: that visual information improves clarity of the paper, leading to more citations and higher impact, or that high-impact papers naturally tend to include new, complex ideas that require visual explanation. Moreover, we find that citations from within the same field tend to correlate with tables while citations from other fields tend to correlate with diagrams, suggesting that visual representations aid interdisciplinary communication.

More broadly, we argue that identification and description of the visual patterns, verified through computational experiments spanning a large corpus of papers, can help improve understanding of how scientific information is best conveyed, how the organization of visual information relates to scientific impact, how best to present scientific information more accessibly to a broader audience, and perhaps most directly, how to build better services for organizing, browsing, and searching the

"visual literature."

Before discussing the above questions, we describe additional steps to deal with data vacancy and estimate the errors made by the figure processing pipeline.

7.1 Data Filtering

In order to conduct a bias-free analysis, we further refine the PMC dataset (Table 3.1) to avoid biases in four ways: First, our analysis of impact depends on having an ALEF score available, so we remove all papers with no ALEF score available (typically because the paper attracted zero citations, and a few negligible cases where processing errors prevented the calculation from completing).

Second, for some papers (less than ten percent of the corpus), the total number of pages could not be determined, preventing us from calculating figure densities. PMC does not report page counts in the XML so we had to determine the page counts using the PDF files provided. However, some papers had no PDF file included, so we could not determine the page count. Third, we remove papers published before 1997 since the number of papers per year from that time is less than 300 and is strongly biased toward a small number of journals that were indexed by PubMed during that period.

Forth, we exclude 86,205 papers with zero figures since we cannot properly distinguish between two situations: (1) papers that were published containing no figures and (2) papers that were published with figures, but for which the figures were not provided to PMC. Generally, more recent papers are more likely to fall into case (1), since the procedure to upload figures separately was more commonly used in the past. Papers corresponding to case (2) (i.e., older papers) can skew the results, since older papers tend to have more citations and therefore higher ALEF scores.

The papers that failed to meet one or more of these criteria appeared to be distributed uniformly across the overall dataset, so any bias created by their removal appears negligible.

After these preprocessing steps, the dataset includes 494,663 papers and 6,897,810 figures (after dismantling), excluding equations. We exclude equations because not all equations were represented as figures and sometimes multiple equations appear in a single figure, making it difficult to

estimate the total number of equations.

Some of the PMC literature is in pre-print formats rather than the official journal format. For these papers, we use the total number of pages from PMC. As a result, the page count may be different than the actual paper. In addition, we underestimate the total number of tables from those authors who use only latex or Microsoft Word to construct their tables, since these authors typically do not provide tables as separate images.

The dataset does not necessarily represent all relevant papers. Authors of the papers analyzed here can voluntarily select to submit papers to PMC, and PMC will clearly tend to attract papers in the life sciences with an emphasis on human biology. In particular, *Nature* publishes a significant number of Physics papers, but these papers will be underrepresented in PMC.

7.2 Visual Patterns Across Disciplines

To analyze the patterns of visual encodings across disciplines, we normalize the individual figure counts by the total number of pages in order to measure the *density* of each figure type. This figure count normalization is similar to the method used by Fawcett et al. [28] in their analysis of equations. It ensures the values are comparable between articles with diverse lengths.

Next we aggregate the figures and papers by journal and research topics to see how figure types vary across publishing venues and disciplines. Figure 7.1 and Figure 7.2 show the average figure density of journals and research topics for which we were able to collect at least 850 or 1000 papers published during 1997 to 2014 from PMC, respectively. Figure topics were assigned used Thomson-Reuters' Journal Citation Report (JCR) category system.

In Figure 7.1, we restricted the analysis to those journals with at least 850 articles in the corpus. The stacked bars present the densities of diagrams, photos, visualizations, and tables, from left to right. Equations are not considered in this case because defining the quantity of equations can be vague: a single image may contain any number of equations, and our dismantler algorithm was not designed to parse equations. The thin dark bars represent the impact of each journal as measured by ArticleInfluence (AI) for the journal [101]. AI is a journal-level metrics, whereas ALEF is an article-level metric.

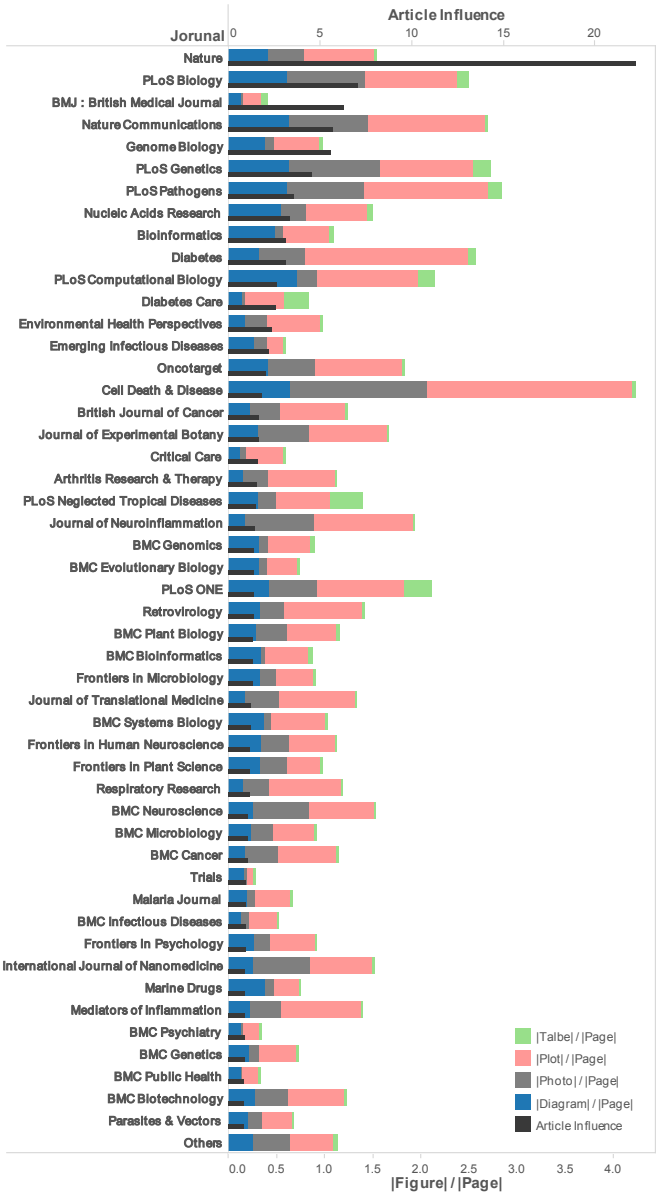

Figure 7.1: The distribution of figure types across journals show an emphasis on plots and diagrams relative to tables, and identify visualization-heavy venues such as Cell Death and Disease. We considered the top 49 highest-impact journals in PMC that had at least 850 papers available in the corpus, where impact is measured as Article Influence (AI) (the black bar). Each stacked bar shows the average density of each figure type across all papers published in the journal. The density of a figure type is the number of instances of that type divided by the page count. The category "Others" contains 288,953 papers from other journals.

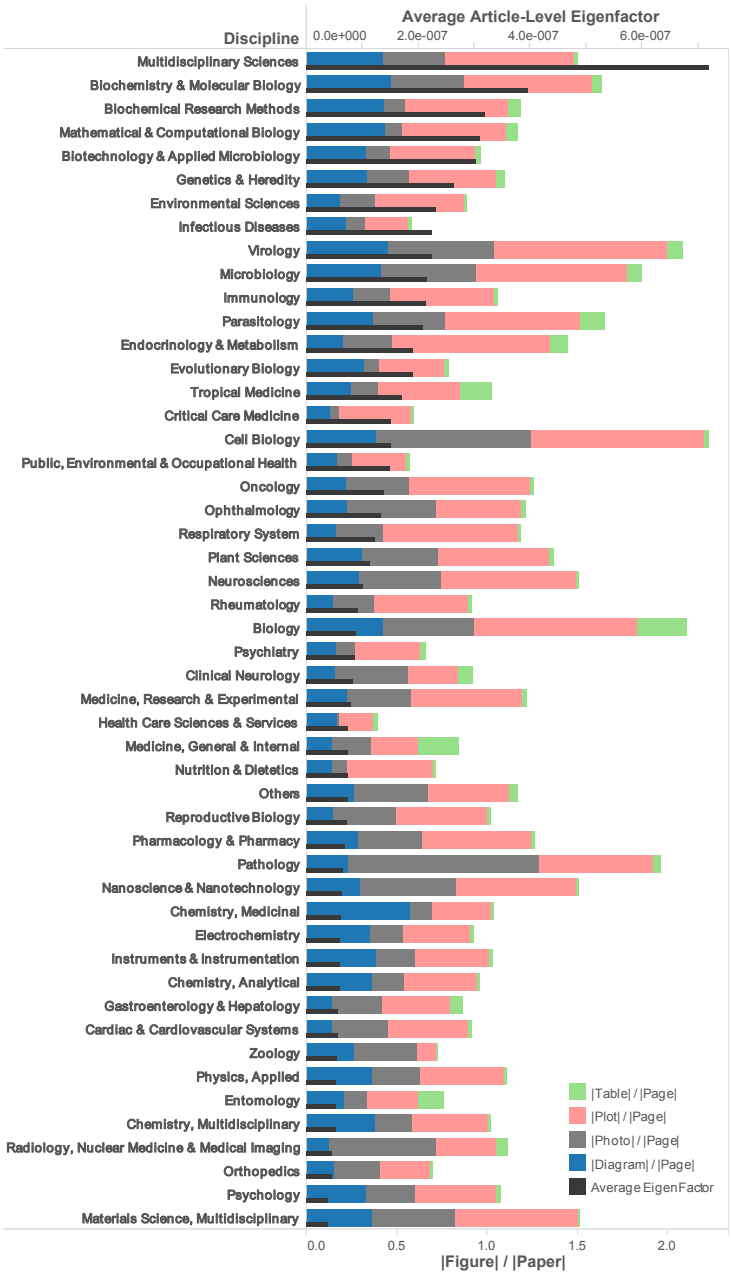

Figure 7.2: Figure distribution by research topic show that microbiology topics tend to emphasize visual presentation of ideas. Topics were determined by the journal categories in Thomson Reuters' JCR. We show the highest-impact 49 topics that have at least 1000 papers, where impact is the average of all papers assigned to that category. The category "Others" includes 216,380 papers from other topics and papers without topic labels.

In Figure 7.2, we used the average ALEF score to estimate the value of topic areas, because topic areas consist of overlapping journals. The AI score is a citation metric for measuring journal influence [101]. The underlying citation data comes from Thomson-Reuters' JCR. Journals and research topics are listed by impact in descending order. Due to the limit of page capacity, we show only the top 49 items and gather the papers from small-collection journals and lower-rank journals into "Others."

Figure 7.1 shows the top 49 journals ordered by AI. Differences exist between journals. The journal *Cell Death and Disease* relies heavily on microscopy and experimental evidence, and we see this emphasis manifest as a significantly higher number photos and plots. We can see that multidisciplinary journals, such as the *Nature* series and the *PLoS* series exhibit a balance of figure types. Qualitatively, many of the journals with high figure-per-page counts are also high in AI. Further, papers from the top one-third journals (16 out of 50) tend to have more diagrams. Journals emphasizing prose-oriented case studies are exceptions and have fewer figures: *British Medical Journal*, *Diabetes Care*, and *Emerging Infectious Diseases*. In comparison, papers from the journals near the tail show lower diagram density. We will make this observation statistically precise in Section 7.4.

Using Thomson Reuters' JCR, we can assign each journal to a research topic, then repeat the analysis of figure distribution by research topic rather than journal. Figure 7.2 shows the disciplines for which at least 1000 papers were available. Differences between disciplines in figure type density are apparent. For example, *Cell Biology* and *Pathology* have a relatively high number of photos per page, whereas *Mathematical and Computational Biology* and *Medicinal Chemistry* have fewer photos per page and relatively more diagrams and plots per page. *Biology* and *Internal Medicine* tend to have relatively more tables per page, suggesting an emphasis on (or tolerance of) presenting quantitative results numerically. We conjecture that these patterns relate cultural norms for publication rather than specific research methods; that certain fields expect a certain "syntax" for a research paper and that the distribution of figures is a part of the syntax. A study of these conjectures is beyond the scope of this paper.

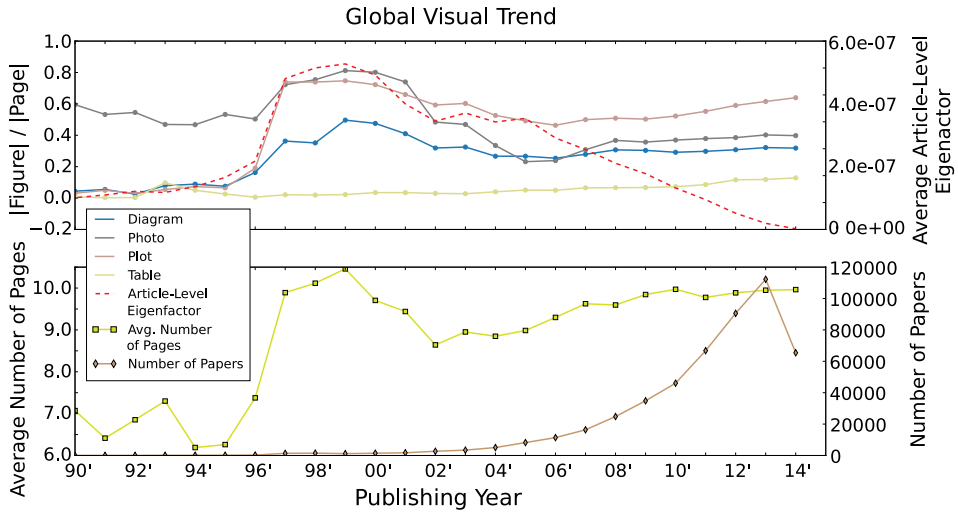

Figure 7.3: The distribution of figure types in the PMC corpus over time. The top figure shows the number of papers increasing dramatically in the mid-2000s, which can be explained by a change in sponsor rules: NIH required authors to submit their papers to PMC. The "hump" of impact between 1997 and 2005 may be attributable to author bias in voluntarily uploading their highest-impact papers. After 2006, the increasing uses of plots and tables may be attributable to increased emphasis on data-intensive research. The density of photos and diagrams are consistently flat over time. The bottom plot provides context: the average page length per paper over time, and the number of papers in the corpus over time.

7.3 Visual Patterns Over Time

We analyze patterns of visual information over time by segmenting the data into different publishing years. The earliest paper we collected from PMC was published in 1937, but relatively few papers earlier than 1997 are included (biasing the corpus). We plot the total number of papers in our database from 1990 to 2014 in Figure 7.3. Paper quantity reaches the thousand mark in 1997 and the ten thousand mark in 2007. In 2008, NIH mandated that authors upload their papers to PMC, partially explaining the growth of the corpus. Papers can be uploaded at any time for any publication year, so we do not necessarily see an increase in later papers. The average ALEF score increases until 2000 and then decreases, consistent with most measures of impact that are inherently time-sensitive.

The "hump" that occurs in Figure 7.3 around 1997 to 2002 is attributable to a bias in the corpus: in this period, the corpus was dominated by just three journals: *Journal of Cell Biology* (38%), *Journal of Experimental Medicine* (31%), and *Journal of General Physiology* (8%) (see Table 8 for details). As more journals were added to PMC, this sampling bias decreased, and the patterns stabilized. After 2006, the number of diagrams per page remains relatively consistent, and a small but consistent growth in the number of plots and tables per page is observed. We conjecture that these increases could be attributable to an increased emphasis on data-intensive science in the biological and biomedical disciplines, but another possibility is that such figures became easier to create thanks to improved tools resulting in increased use.

Table 7.1: Top 3 journals based on quantity for 4 time periods. This table helps explain the patterns over time in Figure 7.3. The numbers in parentheses denote the ranking of the journals in paper quantity. For example, after the year *PLoS One* launched in 2006, this journal becomes the dominant publisher in PubMed Central. It counts for 21% papers from 1997 to 2014 in our database.

Journal	Proportion to All			
	97'-2k	01'-04'	05'-10'	10'-14'
J. Cell Biol	38% (1)	11% (2)	1%	<1%
J. Exp. Med.	31% (2)	10% (3)	<1%	0%
J. Gen. Physiol.	8% (3)	3%	<1%	≪1%
Br. J. Cancer	8%	13% (1)	2%	<1%
PLoS ONE	0%	0%	11% (1)	26% (1)
Nucleic Acids Rs.	0%	0%	4% (2)	<1%
BMC Bioinform.	≪1%	2%	3% (3)	<1%
Sci. Rep.	0%	0%	0%	2% (2)
Int. J. Mol. Sci.	0%	0%	<1%	1% (3)

In Figure 7.4, we select five journals with unique features for closer inspection: *Nature* (highest impact according to our measures), *Cell Death and Disease* (highest figure density), *British Medical Journal* (lowest figure density), *Genome Biology* (unusually low proportion of photos)

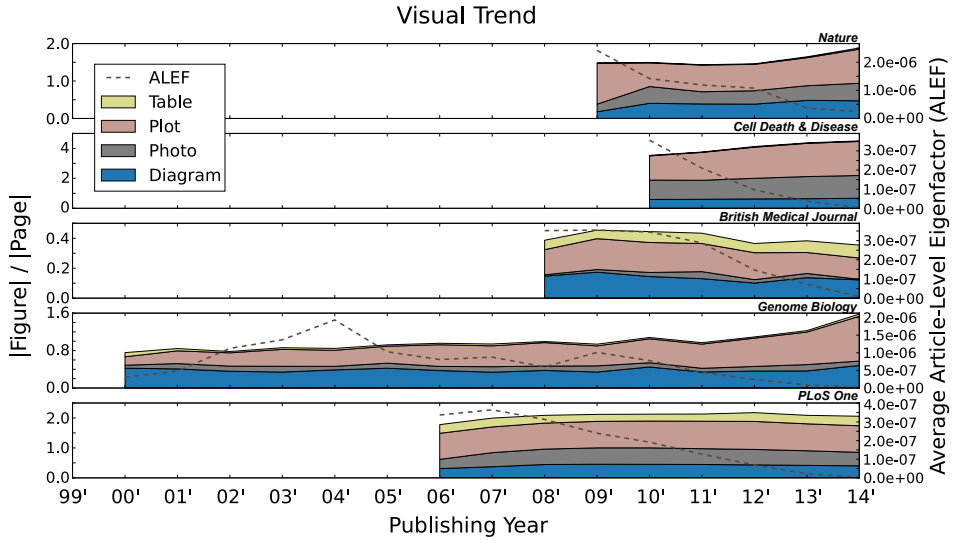

Figure 7.4: We choose five specific journals for closer inspection: *Nature* (highest impact), *Cell Death and Disease* (highest figure density), *British Medical Journal* (lowest figure density), *Genome Biology* (unusually low proportion of photos) and *PLoS One* (largest number of papers). *Nature, Cell Death and Disease* and *Genome Biology* exhibit a recent increase in plots-per-page, consistent with the overall trend. We conjecture that the articles in these high-impact journals are becoming more data-centric. Moreover, *Nature* and especially *Cell Death and Disease* show a heavy use of figures, in part because these journals tend to have greater proportions of multi-chart figures (67% for *Nature* and 82% for *Cell Death and Disease* relative to 30% for the entire image set.) The *British Medical Journal* shows a different trend in which figure density gradually decreases; the mechanism behind this trend is unclear. *PLoS One* shows no significant change from its launch in 2006.

and *PLoS One* (largest number of papers). *Nature* exhibits an increase in figure density over time, driven primarily by an increase in plot density which may reflect an increased emphasis in data-intensive science. For the journal *Cell Death and Disease*, one sees the same effect of growing figure density over time, which corresponds to an increased use of multi-chart figures: 81% of the figures are multi-chart compared to an average of 38%.[1] In contrast, the *British Medical Journal* exhibits low figure density and a gradual decrease in the use of figures over time. Tables are used more in proportion compared to most journals and photos are extremely rare. We conjecture that the decrease in visual information over time may be related to a known shift in focus for BMJ, in

[1]Equations are not taken into account.

which the editor has intentionally focused on topics of broad public interest [72]. It is possible that heavy use of quantitative data in the form of plots may make articles *less* accessible. *Genomics Biology* was selected for its unusually low proportion of photos, which appears consistent over time. We do see the density of plots increasing significantly since 2011, following the global trend. We selected *PLoS One* because of the extremely large number of papers in the corpus. Because it is broadly multidisciplinary, the patterns of figures represent many fields of study and we do not expect, nor do we see, any distinctive pattern. *PLoS One* may represent a microcosm of the overall literature in this regard.

7.4 Visual Patterns Related to Article Influence

Motivated by the increasing need to communicate results across disciplines and with the general public, we study the relationship between the use of figures in the biomedical literature and scientific impact. We hypothesize that an increased use of explanatory figures is associated with increased citations, suggesting that encoding results visually improves communicability. The type of encoded information can be revealed by the type of figure: diagrams to illustrate conceptual ideas or mechanisms, tables and plots to highlight experimental results, and photographs to show clinical evidence. In order to explore the impact of figure type, we further refine the dataset by removing papers with 0 citations. It does not affect the qualitative results, but we find the analysis is simplified when removing this group of papers. These papers are special group, which include papers too recent to be cited, non-biomedical papers, etc. After the refinement, we ended up with 221,199 papers and 1,629,089 images. We verified that our classifiers and dismantler exhibit no bias with respect to impact ensuring that the relationship we see between impact and figure use cannot be explained by biased classification errors or dismantling errors. In addition, we simulated the misclassification to analyze the impact of the uncertainty on correlations.

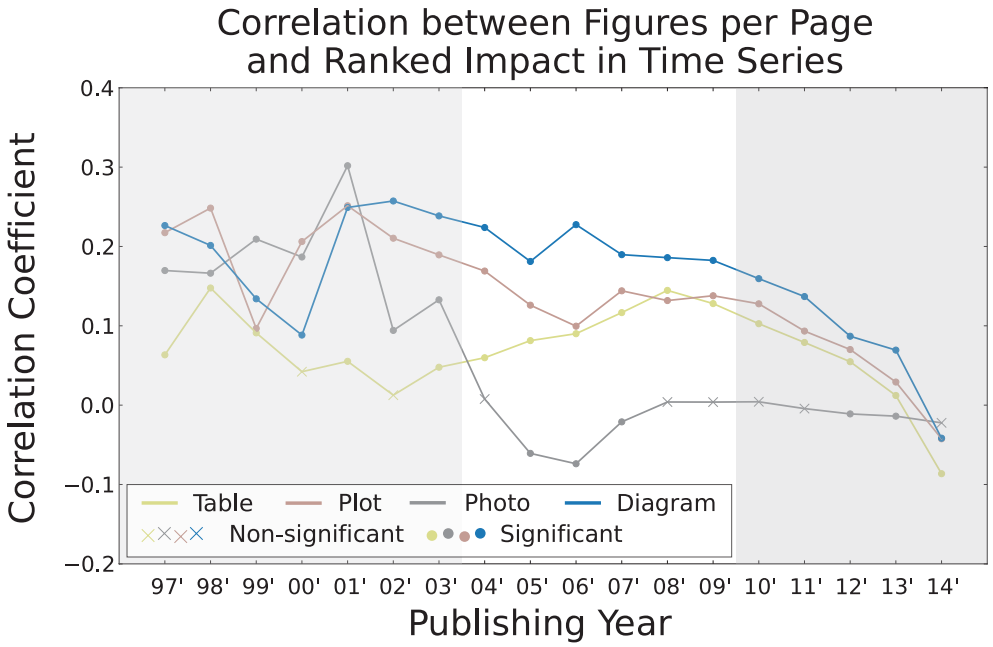

Figure 7.5: **Correlation between Figure Density and Article Impact Over Time.** To avoid age bias that younger papers usually have lower impact, we break down the estimation into different years. The circle makers denote the the results is statistically significant, otherwise the data points are shown by the cross makers. The right shaded area from 2009 to 2014 is considered less credibility because the citation network has well constructed yet. Consistent positive correlation signals are observed from diagram and plot. Increasing positive correlation signals are observed from table. It may suggest an increased use of data-centric methodologies in the life sciences. In the other hand, negative correlations of photo density are found during 2005 to 2007, while positive correlations are found in early years with selection bias.

7.4.1 Figure Use Correlates to Article Influence

We broke down the papers into different years and estimate the rank-order correlations between figure uses and the article impact in each year. We use Spearman's method to account for the non-linearity of the ALEF score. Figure 7.5 shows the correlation coefficients of the four figure types across the overall time period represented in the dataset. Those correlations that are statistically significant (p-value $>$ 0.05) are represented as a circle mark; otherwise we use a cross mark. The papers in the right shaded region (2010-2014) have not had time to accrue citations. The

papers in the left shaded region (1997-2003) are not representative of the overall population; most papers (60%) are from just three journals: *Journal of Cell Biology* (28%), *Journal of Experimental Medicine* (22%), and *British Journal of Cancer* (10%). We focus our analysis on the unshaded region.

In general, diagrams and plots are consistently positively correlated with impact. We consider the interpretation that visual evidence improves clarity of exposition, which in turn leads to higher citations. However, a causal relationship is of course not supported: it is possible that authors are more likely to use visualizations for their highest-impact ideas and experimental results. In previous section, we found that most of the highest-impact journals (including *Nature*, *PLoS Biology*, *Nucleic Acids Research, etc.*) exhibit higher diagram density as well.

Photographs are correlated with high impact in the early, biased portion of the data (left shaded region), but the correlation is insignificant or even negative in more recent years. We conjecture that the negative correlation may suggest that tight page limits associated with high-impact journals may lead authors to sacrifice photographs; photographs are essentially raw data rather than results. On the contrary, increasing trends are observable from table. It may suggest an increased use of data-centric methodologies in the life sciences.

7.4.2 Impact of Different Type of Figure on Communicability Mapped by Citation Count

We first consider a simple model that relates the log of citation count to figure density for each figure type. Figure 7.6 shows figure density against citation count for each figure type. We use citation count as an intuitive proxy for impact; we consider more advanced measures of impact in next. We sorted the 72,643 papers published in the unbiased region (2004-2009) with their ALEF scores and grouped them by every percentile (99%, 98.0%, etc.). We bin the data into percentiles since comparisons between individual papers is not as meaningful without a complete model of impact. Further, visualizing thousands of individual papers makes the patterns more difficult to discern qualitatively. We adjusted the group boundaries to avoid separating papers with identical ALEF scores. Any two papers with ALEF scores within 1E-12 are regarded as having the same score. In these cases, we move the boundary to the next highest threshold.

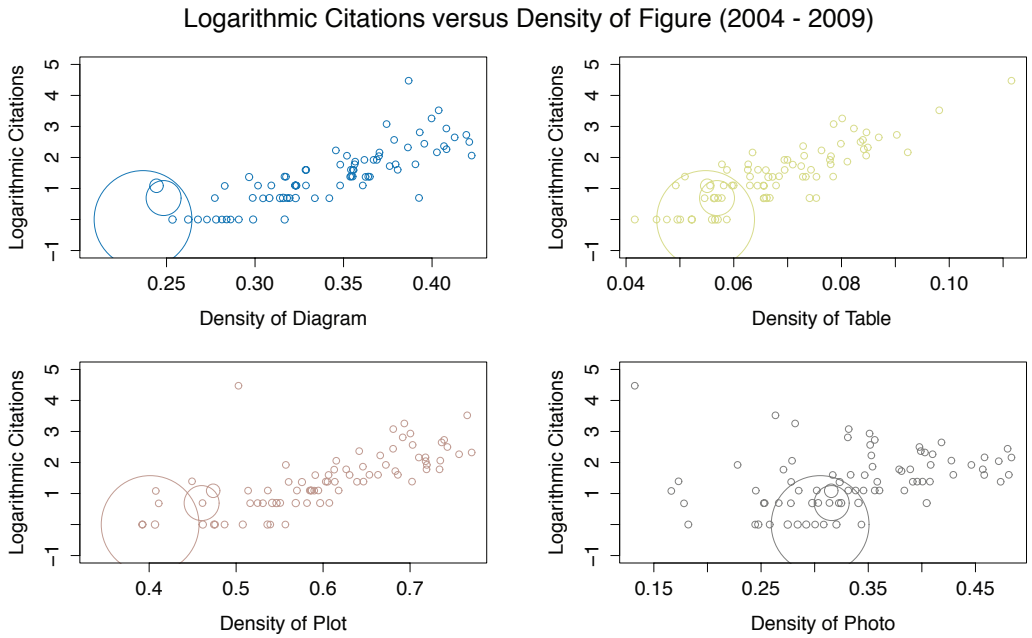

Figure 7.6: **Citations versus Density of Figure.** We zoom into the data from 2004 to 2009 and broke down the sorted papers into groups to avoid plotting 70k dots. Each group counts for 727 papers to achieve a fraction of 1%. The population of a group is visualized by the dot size. Finally 81 groups are produced. The distribution shows the high impact papers tend to use more diagrams, plots and tables. We mixed the papers published in different years in this scatter plot so it blurs the subtle negative correlations of photo found in Figure 7.5.

For each group, we plot the average figure density and average citation count. The size of the group is represented by the size of the mark. The largest mark is the large group of papers that received only one citation.

Figure 7.6 shows that diagrams, tables, and plots are all positively correlated with the log of the citation count, but that photographs are not noticeably correlated. The strong positive correlation of diagrams does not necessarily mean that high-impact papers tend to use more diagrams; in order to compare the impact of different types of figures on article influence, we conducted a regression analysis using negative binomial model [28], specified by the package glm in R MASS library. The negative binomial model is selected instead of a Poisson model, because the citation data is extremely overdispersed. We included the paper age to control for its influence on citation rate.

The number of pages is also included in the model to control the influence of article length. To check the sensitivity of our results to the elimination of papers with zero citation , we also fitted a zero-inflated negative binomial to the unfiltered data. It gave the same statistical conclusions and quantitatively similar estimates of the regression coefficients, so we present only the negative binomial models in this study.

Table 7.2: **Variables affecting the number of citations.** The table shows the mean change in the accrued citations for one standardized unit of change in the independent variables: diagram density, photo density, plot density, table density, and age analyzed by a generalized linear model with a negative binomial error structure [28, 104]. Diagram and Table have the highest Sd. OR among the four figure types. It may indicate that well presenting theories, ideas, or mechanism with the aid of graphics can impact the citations as strong as the providing solid quantitative evidence. The Sd. OR (showing with a 95% CI) indicates the factor change of citations with a standardized unit increase of the independent variable. For instance, each additional standardized diagram per page with OR of 1.11 relates to an 11% increase of citations, whereas an extra standardized photo per page with OR of 0.98 relates a 2% decrease of citations.

	Papers published during 2004-2009		
Parameter	Sd. OR (95% CI)	Z	P
Intercept	1.66 (1.59-1.74)	-23.22	$<$0.001
Density of Diagram	**1.11** (1.10-1.12)	26.70	$<$0.001
Density of Photo	0.98 (0.97-0.98)	-6.04	$<$0.001
Density of Plot	1.06 (1.05-1.07)	14.28	$<$0.001
Density of Table	**1.12** (1.11-1.12)	29.42	$<$0.001
Paper Age	1.19 (1.18-1.20)	45.25	$<$0.001
No. of Page	1.11 (1.10-1.12)	29.00	$<$0.001

For the regression analysis, we used the papers published during 2004-2009 to eliminate selection bias and age bias. The standardized odds ratio enables fair comparison among variables in different units. For example, a unit of standardized diagram density is roughly equivalent to four diagrams in a paper of median length (nine pages). Table 7.2 shows that one standardized unit of

diagram, plot, or table density is associated with 11%, 6%, and 12% **more** citations respectively. In the contrary, a marginal increase in photograph density is associated with 2% fewer citations. The age of the paper has the strongest association with citation count, as expected. Paper age is underestimated in this model because we ignore young papers. Diagram and table density have a stronger association with citation count than plots and photos. We do not propose that additional tables and diagrams will cause higher citations and we do not attempt to predict citations from given figure density. Rather, the regression analysis provides quantitative evidence that figure density can be a factor in scientific impact in the biology and biomedical domains.

7.4.3 Methodology As The Key for Inter-field Citations

Table 7.3: **Variables affecting the number of citations from papers in same fields or different fields.** Table density shows the highest Sd. OR when considering same-field citations; whereas it is taken over by diagram density when considering cross-field citations. It indicates the audiences from the same field may prefer articles with visualized quantitative results but the audiences with diver backgrounds may prefer the articles with visualized conceptual content.

	Intra-field Citations			Inter-field Citations		
Parameter	Sd. OR (95% CI)	*Z*	*P*	Sd. OR (95% CI)	*Z*	*P*
Intercept	0.32 (0.30-0.34)	-34.81	<0.001	1.39 (1.33-1.46)	13.33	<0.001
Den. of Diagram	**1.14** (1.13-1.16)	23.89	<0.001	**1.09** (1.08-1.11)	20.70	<0.001
Den. of Photo	0.95 (0.94-0.96)	-8.30	<0.001	0.99 (0.98-0.99)	-3.08	<0.001
Den. of Plot	**1.09** (1.07-1.10)	14.43	<0.001	**1.05** (1.04-1.06)	10.63	<0.001
Den. of Table	**1.25** (1.23-1.26)	41.41	<0.001	**1.06** (1.05-1.07)	13.49	<0.001
Paper Age	1.29 (1.27-1.30)	44.60	<0.001	1.15 (1.14-1.16)	32.31	<0.001
No. of Page	1.09 (1.08-1.10)	15.43	<0.001	1.12 (1.11-1.13)	27.14	<0.001

We further investigate the relation of figure use on citations from the same field or different field in journal level. We define an *intra-field citation* as a citation from same Thomson-Reuters'

Journal Citation Report (JCR) category as the journal of the cited paper; otherwise, the citation is defined as an inter-field citation. Since the Thomson-Reuters categories are relatively narrow (for example, Virology, Hematology and Oncology are categories), most of a paper's citations are typically inter-field. Table 7.3 shows the impact of the six variables on intra-field inter-field citations separately. Remarkably, table density is far more strongly associated with intra-field citations than with inter-field citations, suggesting that peer researchers in the same domain tend to cite papers due to the raw data or statistical comparisons they contain. Inter-field citations, however, are more strongly associated with diagrams, suggesting that the methods, concepts, and mechanisms are more influential for researchers in other domains. We conjecture that the distribution of visual evidence in a paper could be optimized depending on the audience one wishes to reach.

7.4.4 The Impact of Imperfect Classification on Correlation Results

Since our classifier is imperfect, it is possible that the correlation we measure is an artifact of mistakes in classification. One possibility is that the classifier itself behaves differently with respect to high-impact papers, perhaps making more mistakes due to the styles of figures top journals tend to attract. Another possibility is that the errors in our classifiers just happened to produce an unusually high number of diagrams for a sufficient number of high-impact papers to generate a signal, but an unbiased classifier would have shown no correlation with impact. To test these issues, we generated two new test image sets by randomly sampling our image corpus.

To evaluate multi-chart classifier, we sampled 1000 images that are classified as singleton and 1000 images that are classified as multi-chart by the classifier and manually labeled the images for evaluation purposes. We obtained a precision of 84.6% from multi-chart figures and 87.3% from singleton figures. This result is close to what we find in Table 4.5. We will use this image set to show that the error rate of the multi-chart classifier does not vary with the article impact in the next section.

The figure-type classifier may behave differently on singleton images than on multi-chart images. In particular, we were conscious that incorrectly dismantled multi-charts could be misinterpreted as a set of diagrams, artificially inflating our estimates of diagrams, or even worse inflating

Table 7.4: **Evaluation of figure-type classifier in consideration of false-positive singleton figures.** This table shows the confusion matrix of the five categories. The numbers inside parentheses denote singleton figures, while those outside the parentheses denote the sum of singleton figures and multi-chart figures with single figure types. The multi-chart figures that comprise of two or more types of figure are defined as "composite". These figures are false-positive singleton figures. The "Precision All" considers only singleton figures as true positive and the denominators are 1400. Composite figures and multi-chart visualizations cause the low "Precision All" of diagram and failing of identifying photo arrays as multi-charts results the low precision of photo. About the "Precision Singleton", we eliminate all multi-charts and the values are more comparable with Table 4.5 because singleton figures are the majority of our training set. We use this confusion matrix to derive the possibilities of inflation on the number of figures. These possibilities will be used to calibrate our raw data (see Simulating Figure Counts with Classification Error Rate).

	Equation	**Diagram**	**Photo**	**Table**	**Plot**	**Total**
Equation	1391(1391)	16(16)	12(12)	6(6)	31(31)	1456(1456)
Diagram	4(4)	850(823)	82(72)	48(44)	79(75)	1063(1018)
Photo	2(2)	62(31)	1205(644)	5(2)	37(28)	1311(707)
Table	0(0)	58(58)	0(0)	1265(1263)	9(9)	1332(1330)
Plot	3(2)	331(129)	32(24)	47(36)	1195(1088)	1608(1279)
Composite	0	83	69	29	49	230
Total	1400(1399)	1400(1057)	1400(752)	1400(1351)	1400(1231)	7000(5790)
Precision All	99.4%	58.8%	46.0%	90.2%	77.7%	74.4%
Precision Singleton	99.4%	77.9%	85.6%	93.5%	88.4%	90.0%

more diagrams for high-impact papers. We eliminate this possibility by showing (1) the classification errors are not biased with respect to the impact and (2) the correlation results still stand when we simulate the counts of each figure type up or down by the known error rates for the classifier. We collected a new image set for this experiment. Considering that multi-chart figures (false-positive singletons) can be fed to the figure-type classifier in our pipeline, we randomly sampled 1400 images that are labelled as singleton from each category. Table 7.4 shows the confusion matrix of this image set. The numbers inside parentheses denote singleton figures, while those outside the

parentheses denote the sum of singleton figures and multi-chart figures of the same type (e.g. a multi-chart figure comprising 3 plots). The multi-chart figures that comprise two or more types are defined as "composite". These figures are false positives produced by the multi-chart figure classifier. The table shows the false singletons do inflate the number of figures particularly in diagrams. Precisely, the false diagrams mostly come from multi-plots. The low number of singleton photos is due to the failure of identifying photo arrays as multi-chart figures. These unextracted photos can be regarded as a random noise because the multi-chart classifier is not biased respect to the impact (shown in the next section).

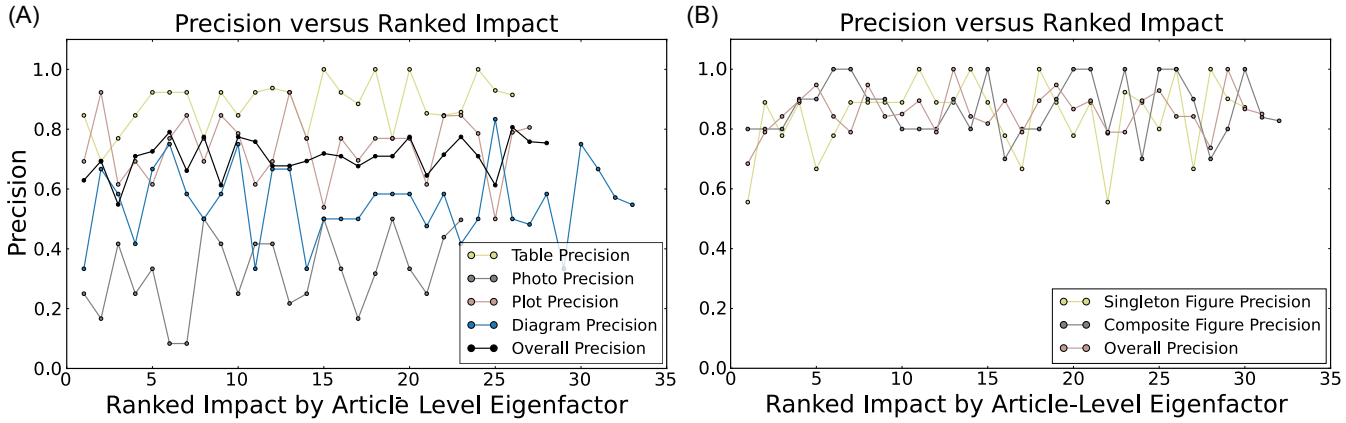

Figure 7.7: **Classification Bias.** (A) We randomly sampled 7000 figures (1400 for each type) that are classified as singleton and manually labelled them. To verify that the misclassification does not correlate to the article impact, we use the bin method introduced previously to group the figures, which has been sorted by their average ALEF scores of their source papers. We filtered figures without available ALEF scores and end up with 6157 figures. Due to the small size of ground-truth data, we set the percentile to 1% to ensure each bin containing 10 figures or more. It shows no correlation between "Precision All" and the average ALEF score. Thus misclassification can be regarded as an unbiased random noise. (B) We randomly sampled 1000 figures that are classified as singleton and another 1000 figures that are classified as multi-chart to estimate the bias of multi-chart figure classifier. We end up with 1790 figures with available ALEF scores. By repeating the same method, it shows no correlation between the precision and the average ALEF score either.

Classification Bias Does Not Explain Correlation Results

If the classifier tended to make more mistakes on higher impact papers, the correlation estimated between article impact and figure density could be explained as an artifact of this bias. We show that the error rate of neither of our two classifiers vary with the article impact. We filtered figures without available ALEF scores[2]; from the image set of testing the multi-chart classifier, we end up with 1790 figures and from the image set of testing the figure-type classifier, we end up with 6157 figures. We use the binning method to group the figures and sort the groups by their average ALEF scores. Due to the small size of ground-truth data, we set the percentile to 1% and it ensures each group containing at least 8 figures. Figure 7.7 shows no correlation found between the precision and the article impact from the two classifiers. Therefore the classification errors can be regarded as an unbiased random noise.

Dismantling Bias Does Not Explain Correlation Results

Since over 60% of figures are embedded in multi-chart figures, our result could be explained by dismantling errors: if for high-impact papers the dismantler is more likely to generate broken fragments that are classified as diagrams, then that would explain our finding. In this section, we show that this explanation does not hold, as the dismantling errors are not biased with respect to the impact (Figure 7.8).

We randomly sampled 500 figures that are classified as multi-chart and review their dismantling results. We manually labelled the sub-images into 8 categories: equation, diagram, photo, table, plot, fragment, multi-chart, and composite, where a fragment is a sub-image containing only missing text. We also manually generated the ground-truth data of ideal dismantling and counted the number of figures in each categories (no fragment, multi-chart, and composite in this case). We use the criterion proposed in our previous work [60] that considers an array of photographic images to be one unit if the author assigns a part label for the array. For the case that the author assigns part

[2]We sampled the images from the full PMC image corpus but not the refined corpus, so some papers have no ALEF scores.

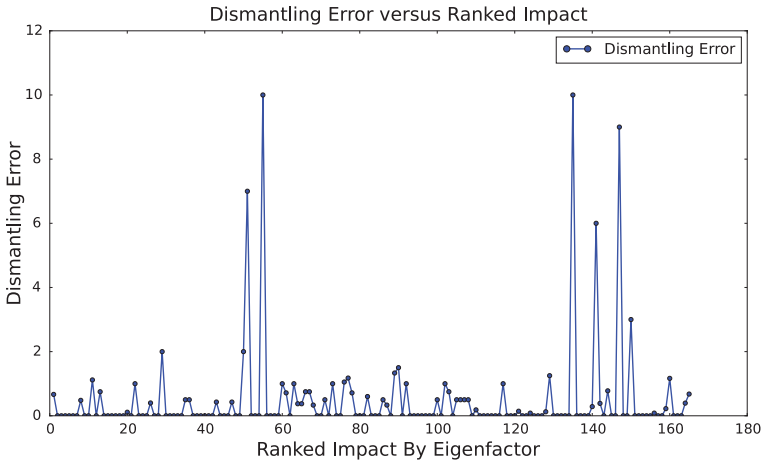

Figure 7.8: **Dismantler Bias.** We randomly sampled 500 figures (eliminated 38 figures with no ALEF scores in this plot) that are classified as multi-chart and compared their segmentation results to the ground-truth data (decomposed by human). We calculated the dismantling errors by calculating L1 norm of correct sub-figures and extracted sub-figures in each category. Then normalized the value to the number of correct sub-figures. It shows no correlation between the dismantling error and ALEF score of the source paper of the figure, eliminating one possible alternative explanation of our correlation result.

labels for every photographic image, we consider them as independent photos to ensure that we do not artificially improve our results. The dismantler correctly extracted 82.9% of the sub-figures from the 500 multi-chart figures and 84.3% of the extracted sub-images are considered correct (not fragments, multi-charts and composites). We obtain a better result compared to our previous work because the testing images used in our previous work are all composite figures (comprising two or more types of figure). Correctly decomposing composite figures is usually more difficult than decomposing multi-chart figures with single figure type due to the higher possibility of unorganized layout found in composite figures. Figure 7.8 shows no correlation between dismantling error and article impact, where the dismantling error is mapped by

$$\frac{\sum_{i \in \text{categories}} \left| N_i^{\text{correct sub-figures}} - N_i^{\text{extracted sub-figures}} \right|}{\sum_{i \in \text{categories}} \left| N_i^{\text{correct sub-figures}} \right|}$$

, where N denotes the number of sub-figures. We grouped the images from the papers with

the same ALEF scores and reported the mean value of the dismantling error for the group with two or more images. Thirty eight multi-chart images with no ALEF score are eliminated from this estimation.

Simulating Figure Counts with Classification Error Rate

In this section, we determine that the detected correlation between the article impact and figure use cannot be explained as a side effect of the errors of our classifier.

To estimate for this error, we can adjust the estimated counts of each figure type up or down by the known error rates for the classifier. For example, if the classifier is known to misidentify a visualization as a diagram 10% of the time, then we should adjust the estimated number of diagrams down by 10% and the estimated number of visualizations up by 10%. However, this correction assumes that the errors are fixed; it is still possible that our classifier made an unusually bad guess and mislabeled many images as diagrams, generating a false signal. To measure the likelihood of this case, we ran a series of 1000 experiments in which we randomly assigned a label based on the confusion matrix of the classifier. For example, a figure that was originally labelled as a diagram may be relabeled as as an equation with a probability of 1.5% (16/1057); a photo, 2.9% (31/1057); a table, 5.5% (58/1057); or a visualization, 12.2% (129/1057). Otherwise, it remains a diagram. By shuffling the labels this way, we can determine the sample distribution of our noisy classifier. If the resulting distribution contains zero correlation within a 95% confidence interval, then the signal we detected can be explained as a side effect of the errors of our classifier.

Figure 7.9 shows the distribution of the correlation coefficients from the group of papers published in 2008 from the 1000 trials. The distributions for diagrams (blue), plots (red), and tables (yellow) are still significantly above zero after the simulation. The dotted lines are the correlation coefficients obtained from the raw data without adjustment applied. The peak shift depicts the adjustment effect and the peak width indicates the interval of possible correlation coefficients. From this experiment, we obtained remarkable statistically significant correlations for diagram (0.175643 +/- 0.000204), plot (0.140698 +/- 0.000119) and table (0.157066 +/- 0.00248). The fail rate of 0.12 from the photo means that the correlations from 120 trails are not statistical significant.

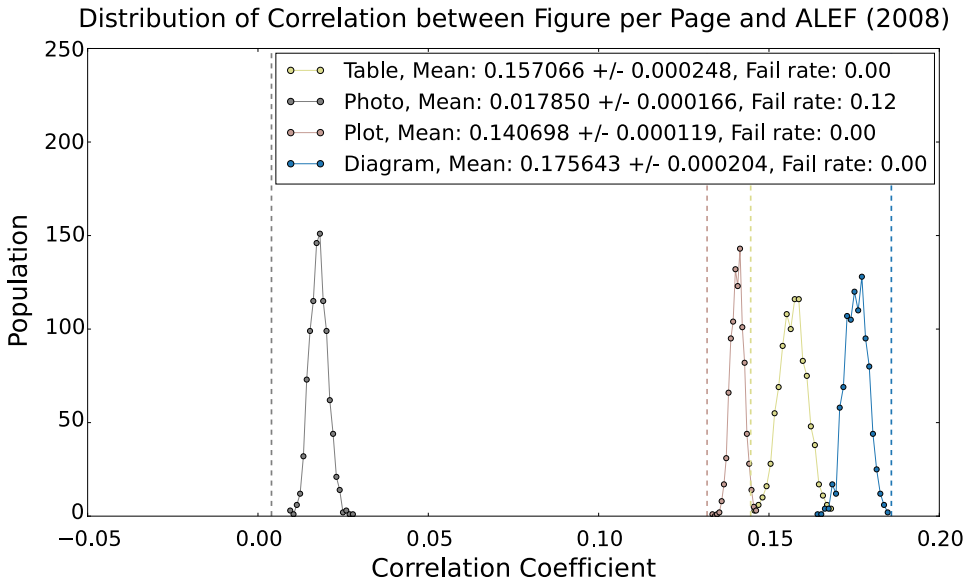

Figure 7.9: **Simulation of correlation coefficients for papers published in 2008.** Considering that the machine-labels can be mistaken, we simulated the number of figures in each figure type by shuffling the machine labels according to the probabilities derived from Table 7.4. We ran 1000 trials to and plotted the distribution of the correlation coefficients. The dotted lines are the correlation coefficients obtained from the raw data without adjustment. The fail rate means the proportion of simulations that produce non-significant correlations. The peak shift depicts the adjustment effect and the peak width indicates the interval of possible correlation coefficients. From the simulations, we obtained statistically significant correlations for diagram (0.175643 +/- 0.000204), plot (0.140698 +/- 0.000119) and table (0.157066 +/- 0.00248).

7.5 Summary

In this chapter, we present the visual patterns of scientific literature. In different disciplines, we found that the role of the five figure types can vary widely. For instance, clinical papers tend to have higher photo density and computational papers tend to have higher diagram and plot density. In respect to visual patterns over time, we found a growing use of plots, perhaps suggesting increasing emphasis on data-intensive methods. When taking the article influence into the analysis, we found that the number of diagrams, plots and tables per page correlates positively with impact, while the number of photographs are slightly negatively correlated. Moreover, we find that citations within

the same field tend to correlate with tables while citations from other fields tend to correlate with diagrams, suggesting that visual representations aid interdisciplinary communication.

Chapter 8

CONCLUSION

In this study, we aim to facilitate a variety of research projects over scientific figures, an area we call viziometrics. It extends prior work in bibliometrics and scientometrics but focuses on the role of visual information encoding. We developed a figure processing pipeline that automatically classifies figures into equations, diagrams, plots, photos, tables, and more sub-types of the above. To facilitate further research on these visual objects, we release both the code and the data for other researchers to explore. By integrating the figure-type labels and article metadata, we found that the number of diagrams and plots per page correlates positively with article influence. We consider the interpretation that visual evidence improves clarity of exposition, which in turn leads to higher citations. Moreover, we find that citations from within the same field tend to correlate with tables while citations from other fields tend to correlate with diagrams, suggesting that visual representations aid interdisciplinary communication. Based on this finding, we have presented VizioMetrics.org, a platform for mining millions of figures from the biomedical sciences. Our hope is that the platform will catalyze future research for improving scholarly search and facilitating large-scale analysis of these figures and new figure-centric applications. VizioMetrics.org provides a figure-oriented search service for general academic users and an open data resource for researches interested in mining scholarly figures. We also develop a crowdsourced labelling application for further labeling and subsequent improvement of our machine learning methods and methods of others. This platform is needed since mechanical turkers do not usually have the domain knowledge for labeling professional figures (e.g., different protein gels, phylogenetic trees,etc). In addition, we presented PhyloParser, a framework that automatically identifies phylogenetic tree figures from scientific literature, extracts the key components of tree structure, and reconstructs them to recover the raw data of species relationships. PhyloParser enables extraction of phylogenies from a large scale of den-

drograms, as a technical foundation to to construct a database of species relationships automatically from the scientific literature, validate these relationships against manually constructed databases, and answer questions about the coverage and veracity of the results, and how the confidence of scientific results has changed over time. We hope the viziometrics.org reduces the activation energy needed to analyze scholarly figures at scale and provokes new and exciting questions. We also encourage people to use our publicly available corpus and software to explore this area of research and create a new community of interest.

BIBLIOGRAPHY

[1] Zanran. http://www.Zanran.com/, 2006.

[2] D8taplex. http://d8taplex.com/, 2011.

[3] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749, 2005.

[4] Amr Ahmed, Andrew Arnold, Luis Pedro Coelho, Joshua Kangas, Abdul Saboor Sheikh, Eric Xing, William Cohen, and Robert F. Murphy. Structured literature image finder: Parsing text and figures in biomedical literature. *Journal of Web Semantics*, 8:151–154, 2010.

[5] Amr Ahmed, Eric P Xing, William W Cohen, and Robert F Murphy. Structured correspondence topic models for mining captioned figures in biological literature. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 39–48. ACM, 2009.

[6] Shaaron Ainsworth and Andrea Th Loizou. The effects of self-explaining when learning with text or diagrams. *Cognitive science*, 27(4):669–681, 2003.

[7] Carl T Bergstrom, Jevin D West, and Marc A Wiseman. The eigenfactor? metrics. *The Journal of Neuroscience*, 28(45):11433–11434, 2008.

[8] Sumit Bhatia, Prasenjit Mitra, and C Lee Giles. Finding algorithms in scientific articles. In *Proceedings of the 19th international conference on World wide web*, pages 1061–1062. ACM, 2010.

[9] Nathalie Boone, Béatrice Loriod, Aurélie Bergon, Oualid Sbai, Christine Formisano-Tréziny, Jean Gabert, Michel Khrestchatisky, Catherine Nguyen, François Féron, Felicia B. Axelrod, and El Chérif Ibrahim. Olfactory stem cells, a new cellular model for studying molecular mechanisms underlying familial dysautonomia. *PLoS ONE*, 5, 2010.

[10] Nathalie Boone, Béatrice Loriod, Aurélie Bergon, Oualid Sbai, Christine Formisano-Tréziny, Jean Gabert, Michel Khrestchatisky, Catherine Nguyen, François Féron, Felicia B Axelrod, et al. Olfactory stem cells, a new cellular model for studying molecular mechanisms underlying familial dysautonomia. *PLoS One*, 5(12):e15590, 2010.

[11] Vicente Botella-Soler, Mario Valderrama, Benoît Crépon, Vincent Navarro, and Michel Le van Quyen. Large-scale cortical dynamics of sleep slow waves. *PLoS ONE*, 7, 2012.

[12] Arthur Brady and Steven Salzberg. Phymmbl expanded: confidence scores, custom databases, parallelization and more. *Nature methods*, 8(5):367–367, 2011.

[13] William Browuer, Saurabh Kataria, Sujatha Das, Prasenjit Mitra, and C Lee Giles. Segregating and extracting overlapping data points in two-dimensional plots. In *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, pages 276–279. ACM, 2008.

[14] Guillaume Cabanac, Gilles Hubert, and James Hartley. Solo versus collaborative writing: Discrepancies in the use of tables and graphs in academic articles. *Journal of the Association for Information Science and Technology*, 65(4):812–820, 2014.

[15] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[16] Zhe Chen, Michael Cafarella, and Eytan Adar. Diagramflyer: A search engine for data-driven diagrams. In *Proceedings of the 24th International Conference on World Wide Web Companion*, WWW '15 Companion, pages 183–186, Republic and Canton of Geneva, Switzerland, 2015. International World Wide Web Conferences Steering Committee.

[17] Beibei Cheng, Sameer Antani, R Joe Stanley, and George R Thoma. Automatic segmentation of subfigure image panels for multimodal biomedical document retrieval. In *IS&T/SPIE Electronic Imaging*, pages 78740Z–78740Z. International Society for Optics and Photonics, 2011.

[18] PM Sagnik Choudhury, Shuting Wang, and L Giles. Automated data extraction from scholarly line graphs. In *GREC*, 2015.

[19] Sagnik Ray Choudhury, Pinaki Mitra, Andi Kirk, Silvia Szep, Donald Pellegrino, Simon Jones, and C Lee Giles. Figure metadata extraction from digital documents. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 135–139. IEEE, 2013.

[20] William S Cleveland. Graphs in scientific publications. *The American Statistician*, 38(4):261–269, 1984.

[21] Adam Coates, Andrew Y Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *International conference on artificial intelligence and statistics*, pages 215–223, 2011.

[22] Tamsyn M Crowley, Volker R Haring, Simon Burggraaf, and Robert J Moore. Application of chicken microarrays for gene expression analysis in other avian species. *BMC genomics*, 10(2):S3, 2009.

[23] Adish Dani, Bo Huang, Joseph Bergan, Catherine Dulac, and Xiaowei Zhuang. Superresolution imaging of chemical synapses in the brain. *Neuron*, 68(5):843 – 856, 2010.

[24] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z Wang. Image Retrieval : Ideas , Influences, and Trends of the New Age. *ACM Computing Surveys*, pages 1–35, 2006.

[25] Derek J. de Solla Price. Networks of scientific papers. *Science*, 149(3683):510–515, 1965.

[26] Hazar Dib, Nicoletta Adamo-Villani, and Stephen Garver. Realistic versus schematic interactive visualizations for learning surveying practices: A comparative study. *Int. J. Inf. Commun. Technol. Educ.*, 10(2):62–74, April 2014.

[27] Per GP Ericson, Ida Envall, Martin Irestedt, and Janette A Norman. Inter-familial relationships of the shorebirds (aves: Charadriiformes) based on nuclear dna sequence data. *BMC Evolutionary Biology*, 3(1):16, 2003.

[28] Tim W Fawcett and Andrew D Higginson. Heavy use of equations impedes communication among biologists. *Proceedings of the National Academy of Sciences*, 109(29):11735–11739, 2012.

[29] Li Fei-Fei and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 524–531. IEEE, 2005.

[30] R.P. Futrelle, I.A. Kakadiaris, J. Alexander, C.M. Carriero, N. Nikolakis, and J.M. Futrelle. Understanding diagrams in technical documents. *Computer*, 25, 1992.

[31] R.P. Futrelle, Mingyan Shao, C. Cieslik, and A.E. Grimes. Extraction,layout analysis and classification of diagrams in pdf documents. In *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*, pages 1007–1013, Aug 2003.

[32] Eugene Garfield. The History and Meaning of the Journal Impact Factor. *JAMA*, 295(1):90–93, 2006.

[33] Michelle G. Gibbons. Reassessing discovery: Rosalind Franklin, scientific visualization, and the structure of DNA. 79(1):63–80, January 2012.

[34] Maxim B Gongalsky, Alexander Kharin, Liubov A Osminkina, Victor Timoshenko, Jinyoung Jeong, Han Lee, and Bong Chung. Enhanced photoluminescence of porous silicon nanoparticles coated by bioresorbable polymers, 2012.

[35] Bryan T Grenfell, Oliver G Pybus, Julia R Gog, James LN Wood, Janet M Daly, Jenny A Mumford, and Edward C Holmes. Unifying the epidemiological and evolutionary dynamics of pathogens. *science*, 303(5656):327–332, 2004.

[36] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.

[37] Melanie L Hand, Noel OI Cogan, Alan V Stewart, and John W Forster. Evolutionary history of tall fescue morphotypes inferred from molecular phylogenetics of the lolium-festuca species complex. *BMC evolutionary biology*, 10(1):303, 2010.

[38] James Hartley and Guillaume Cabanac. Do men and women differ in their use of tables and graphs in academic publications? *Scientometrics*, 98(2):1161–1172, 2014.

[39] 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*, pages 770–778, 2016.

[40] Peter Hegarty and Zoe Walton. The consequences of predicting scientific impact in psychology using journal impact factors. *Perspectives on Psychological Science*, 7(1):72–78, 2012.

[41] Jody Hey. Isolation with migration models for more than two populations. *Molecular biology and evolution*, 27(4):905–920, 2010.

[42] Kyung-Won Hong, Sanghoon Moon, Young Kim, Yun Kim, Dong-Joon Kim, Cheong-sik Kim, Sung Kim, and Bong-Jo Kim. Association between the ABO locus and hematological traits in Korean, 2012.

[43] Weihua Huang and Chew Lim Tan. A System for Understanding Imaged Infographics and Its Applications. In *DOCENG'07: PROCEEDINGS OF THE 2007 ACM SYMPOSIUM ON DOCUMENT ENGINEERING*, pages 9–18, 2007.

[44] Weihua Huang, ChewLim Tan, and WeeKheng Leow. Model-based chart image recognition. In Josep Llads and Young-Bin Kwon, editors, *Graphics Recognition. Recent Advances and Perspectives*, volume 3088 of *Lecture Notes in Computer Science*, pages 87–99. Springer Berlin Heidelberg, 2004.

[45] Joseph Hughes. Treeripper web application: towards a fully automated optical tree recognition software. *BMC bioinformatics*, 12(1):178, 2011.

[46] J.D. West, I. Wesley-Smith, and C.T. Bergstrom. A recommendation system based on hierarchical clustering of an article-level citation network. *IEEE Transactions on Big Data*, 2(2):113–123, June 2016.

[47] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[48] Jianyu Jin, Wei Jing, Xin-Xiang Lei, Chen Feng, Shuping Peng, Kathleen Boris-Lawrie, and Yingqun Huang. Evidence that Lin28 stimulates translation by recruiting RNA helicase A to polysomes. *Nucleic acids research*, 39:3724–3734, 2011.

[49] P. Kalas, J. R. Graham, E. Chiang, M. P. Fitzgerald, M. Clampin, E. S. Kite, K. Stapelfeldt, C. Marois, and J. Krist. Optical Images of an Exosolar Planet 25 Light-Years from Earth. *Science*, 322:1345–, November 2008.

[50] Marina A. Kapina, Galina S. Shepelkova, Vadim G. Avdeenko, Anna N. Guseva, Tatiana K. Kondratieva, Vladimir V. Evstifeev, and Alexander S. Apt. Interleukin-11 drives early lung inflammation during mycobacterium tuberculosis infection in genetically susceptible mice. *PLoS ONE*, 6, 2011.

[51] Saurabh Kataria, William Browuer, Prasenjit Mitra, and C Lee Giles. Automatic extraction of data points and text blocks from 2-dimensional plots in digital documents. In *AAAI*, volume 8, pages 1169–1174, 2008.

[52] Manolis Kellis, Nick Patterson, Matthew Endrizzi, Bruce Birren, and Eric S Lander. Sequencing and comparison of yeast species to identify genes and regulatory elements. *Nature*, 423(6937):241–254, 2003.

[53] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European Conference on Computer Vision*, pages 235–251. Springer, 2016.

[54] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[55] Renaud Lambiotte and Martin Rosvall. Ranking and clustering of nodes in networks with smart teleportation. *Physical Review E*, 85(5):056107, 2012.

[56] Jill H Larkin and Herbert A Simon. Why a diagram is (sometimes) worth ten thousand words. *Cognitive science*, 11(1):65–100, 1987.

[57] Thomas Laubach and Arndt Von Haeseler. Treesnatcher: coding trees from images. *Bioinformatics*, 23(24):3384–3385, 2007.

[58] Thomas Laubach, Arndt von Haeseler, and Martin J Lercher. Treesnatcher plus: capturing phylogenetic trees from images. *BMC bioinformatics*, 13(1):110, 2012.

[59] Po-Shen Lee, Jevin D , West, and Bill Howe. Use of figures correlated with higher impact in the biomedical literature. *In Prep.*, 2016.

[60] Po-shen Lee and Bill Howe. Dismantling composite visualizations in the scientific literature. In *International Conference on Pattern Recognition Applications and Methods, ICPRAM, Lisbon, Portugal*, 2015.

[61] Po-shen Lee, Jevin D West, and Bill Howe. Viziometrix: A platform for analyzing the visual information in big scholarly data. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 413–418. International World Wide Web Conferences Steering Committee, 2016.

[62] Michael S Lew. Content-Based Multimedia Information Retrieval : State of the Art and Challenges. *ACM Transactions on Multimedia Computing, Communications and Applications*, 2:1–19, 2006.

[63] Duncan Lindsey. *The scientific publication system in social science*. Jossey-Bass Inc Pub, 1978.

[64] Xiaonan Lu, J Wang, Prasenjit Mitra, and C Lee Giles. Automatic extraction of data from 2-d plots in documents. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 1, pages 188–192. IEEE, 2007.

[65] Xiaonan Lu, J.Z. Wang, P. Mitra, and C.L. Giles. Automatic extraction of data from 2-d plots in documents. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 1, pages 188–192, Sept 2007.

[66] Virginia Morell. Treebase: the roots of phylogeny. *Science*, 273(5275):569, 1996.

[67] Robert F Murphy, Meel Velliste, Jie Yao, and Gregory Porreca. Searching online journals for fluorescence microscope images depicting protein subcellular location patterns. In *Bioinformatics and Bioengineering Conference, 2001. Proceedings of the IEEE 2nd International Symposium on*, pages 119–128. IEEE, 2001.

[68] Jörg Neddens and Andrés Buonanno. Expression of the neuregulin receptor ErbB4 in the brain of the rhesus monkey (Macaca mulatta). *PLoS ONE*, 6, 2011.

[69] Douglas L Nelson, Valerie S Reed, and John R Walling. Pictorial superiority effect. *Journal of Experimental Psychology: Human Learning and Memory*, 2(5):523, 1976.

[70] Maureen A O?Leary and Seth Kaufman. Morphobank: phylophenomics in the ?cloud? *Cladistics*, 27(5):529–537, 2011.

[71] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[72] Mark Peplow. "no time for stodgy: Crusading editor of the bmj aims to shake things up.". `http://www.statnews.com/2016/01/04/bmj-editor-fiona-godlee/`, 2016.

[73] Rodolfo Pérez-Rodríguez, Omar Domínguez-Domínguez, Gerardo Pérez Ponce de León, and Ignacio Doadrio. Phylogenetic relationships and biogeography of the genus algansea girard (cypriniformes: Cyprinidae) of central mexico inferred from molecular data. *BMC evolutionary biology*, 9(1):223, 2009.

[74] Russell A Poldrack and Tal Yarkoni. From brain maps to cognitive ontologies: Informatics and the search for mental structure. *Annual review of psychology*, 67:587–612, 2016.

[75] V.S.N. Prasad, B. Siddiquie, J. Golbeck, and L.S. Davis. Classifying Computer Generated Charts. *2007 International Workshop on Content-Based Multimedia Indexing*, 2007.

[76] A Rambaut. Treethief: a tool for manual phylogenetic tree entry. *Program distributed by the author. http://evolve. zoo. ox. ac. uk/software/TreeThief/main. html*, 2000.

[77] Sagnik Ray Choudhury and Clyde Lee Giles. An architecture for information extraction from figures in digital libraries. In *Proceedings of the 24th International Conference on World Wide Web Companion*, pages 667–672. International World Wide Web Conferences Steering Committee, 2015.

[78] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en.

[79] Sara Rodriguez-Jato, Robert D Nicholls, Daniel J Driscoll, and Thomas P Yang. Characterization of cis- and trans-acting elements in the imprinted human SNURF-SNRPN locus. *Nucleic acids research*, 33:4740–4753, 2005.

[80] M. Rosvall and C. T. Bergstrom. Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems. *PLoS One*, 6:e18209, 2011.

[81] KC Santosh, Zhiyun Xue, Sameer Antani, and George Thoma. Nlm at imageclef 2015: Biomedical multipanel figure separation. *Working Notes of CLEF*, 2015, 2015.

[82] Manolis Savva, Nicholas Kong, Arti Chhajta, Li Fei-Fei, Maneesh Agrawala, and Jeffrey Heer. ReVision: Automated Classification, Analysis and Redesign of Chart Images. In *UIST '11*, pages 393–402, 2011.

[83] Katharina Scheiter, Peter Gerjets, Thomas Huk, Birgit Imhof, and Yvonne Kammerer. The effects of realism in learning with dynamic visualizations. *Learning and Instruction*, 19(6):481 – 494, 2009.

[84] Mingyan Shao and RobertP. Futrelle. Recognition and classification of figures in pdf documents. In Wenyin Liu and Josep Llads, editors, *Graphics Recognition. Ten Years Review and Future Perspectives*, volume 3926 of *Lecture Notes in Computer Science*, pages 231–242. Springer Berlin Heidelberg, 2006.

[85] Noah Siegel, Zachary Horvitz, Roie Levin, Santosh Divvala, and Ali Farhadi. Figureseer: Parsing result-figures in research papers. In *European Conference on Computer Vision*, pages 664–680. Springer, 2016.

[86] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[87] A W M Smeulders, M Worring, S Santini, A Gupta, and R Jain. Content-based image retrieval at the end of the early years. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22:1349–1380, 2000.

[88] Ray Smith. An overview of the tesseract ocr engine. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 2, pages 629–633. IEEE, 2007.

[89] Srinivasa Subramaniam, Ute Zirrgiebel, Oliver von Bohlen Und Halbach, Jens Strelau, Christine Laliberté, David R Kaplan, and Klaus Unsicker. ERK activation promotes neuronal degeneration predominantly through plasma membrane damage and independently of caspase-3. *The Journal of cell biology*, 165:357–369, 2004.

[90] Neetu Kumra Taneja, Sakshi Dhingra, Aditya Mittal, Mohit Naresh, and Jaya Sivaswami Tyagi. Mycobacterium Tuberculosis Transcriptional Adaptation, Growth Arrest and Dormancy Phenotype Development is Triggered by Vitamin C. *PLoS ONE*, 5, 2010.

[91] Małgorzata Tartanus, Agnieszka Wnuk, Marcin Kozak, and James Hartley. Graphs and prestige in agricultural journals. *Journal of the American Society for Information Science and Technology*, 64(9):1946–1950, 2013.

[92] Mario Taschwer and Oge Marques. Automatic separation of compound figures in scientific articles. *CoRR*, abs/1606.01021, 2016.

[93] Sandeep Tata and Jignesh M Patel. Estimating the selectivity of tf-idf based cosine similarity predicates. *ACM Sigmod Record*, 36(2):7–12, 2007.

[94] Michinori Toriyama, Tadayuki Shimada, Ki Bum Kim, Mari Mitsuba, Eiko Nomura, Kazuhiro Katsuta, Yuichi Sakumura, Peter Roepstorff, and Naoyuki Inagaki. Shootin1: A protein involved in the organization of an asymmetric signal for neuronal polarization. *The Journal of cell biology*, 175:147–157, 2006.

[95] Seval Türkmen, Gao Guo, Masoud Garshasbi, Katrin Hoffmann, Amjad J. Alshalah, Claudia Mischung, Andreas Kuss, Nicholas Humphrey, Stefan Mundlos, and Peter N. Robinson. CA8 mutations cause a novel syndrome characterized by ataxia and mild mental retardation with predisposition to quadrupedal gait. *PLoS Genetics*, 5, 2009.

[96] Petr Čech, Jaromír Kukal, Jií Černý, Bohdan Schneider, and Daniel Svozil. Automatic workflow for the classification of local DNA conformations. *BMC bioinformatics*, 14:205, 2013.

[97] Xiaolong Wang, H Shatkay, and C Kambhamettu. Cis udel working notes on image-clef 2015: Compound figure detection task. *Working Notes of CLEF*, 2015, 2015.

[98] Colin Ware. *Information visualization: perception for design*. Elsevier, 2012.

[99] Ian Wesley-Smith, Carl T Bergstrom, and Jevin D West. Static ranking of scholarly papers using article-level eigenfactor (alef). 9th ACM International Conference on Web Search and Data Mining. ACM, in press.

[100] JD West, M , D Vilhena, and CT Bergstrom. Ranking and mapping article-level citation networks. *In Prep.*, 2016.

[101] Jevin D West, Theodore C Bergstrom, and Carl T Bergstrom. The eigenfactor metricstm: A network approach to assessing scholarly journals. *College & Research Libraries*, 71(3):236–244, 2010.

[102] Jevin D West, Michael C Jensen, Ralph J Dandrea, Gregory J Gordon, and Carl T Bergstrom. Author-level eigenfactor metrics: Evaluating the influence of authors, institutions, and countries within the social science research network community. *Journal of the American Society for Information Science and Technology*, 64(4):787–801, 2013.

[103] Jevin D West, Ian Wesley-Smith, and Carl T Bergstrom. A recommendation system based on hierarchical clustering of an article-level citation network. *IEEE Transactions on Big Data*, 2(2):113–123, 2016.

[104] Gary C White and Robert E Bennetts. Analysis of frequency count data using the negative binomial distribution. *Ecology*, 77(8):2549–2557, 1996.

[105] Ryan P Womack. Research data in core journals in biology, chemistry, mathematics, and physics. *PloS one*, 10(12):e0143460, 2015.

[106] Ziheng Yang and Bruce Rannala. Molecular phylogenetics: principles and practice. *Nature Reviews Genetics*, 13(5):303–314, 2012.

[107] Naoko Yokokura and Toyohide Watanabe. Layout-based approach for extracting constructive elements of bar-charts. In Karl Tombre and AtulK. Chhabra, editors, *Graphics Recognition Algorithms and Systems*, volume 1389 of *Lecture Notes in Computer Science*, pages 163–174. Springer Berlin Heidelberg, 1998.

[108] Kaizhong Zhang and Dennis Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing*, 18(6):1245–1262, 1989.

[109] Yan Ping Zhou Yan Ping Zhou and Chew Lim Tan Chew Lim Tan. Hough technique for bar charts detection and recognition in document images. *Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101)*, 2, 2000.

[110] Yanping Zhou and Chew Lim Tan. Learning-based scientific chart recognition. In *4th IAPR International Workshop on Graphics Recognition, GREC2001*, pages 482–492, 2001.

[111] Daniel Zwillinger and Stephen Kokoska. *CRC standard probability and statistics tables and formulae*. CRC Press, 1999.

Appendix A

WHERE TO FIND THE DATA AND SOURCE CODE

Viziometrics dataset is open access for academic use and available via the APIs.

- VizioMetrics.org

  `http://www.viziometrics.org`

- VizioMetrics Open Access APIs

  `http://www.viziometrics.org/API`

- My GitHub

  `https://github.com/sephonlee/`

Appendix B

COPYRIGHT AND ATTRIBUTION