Shop Signboards Detection and Classification Framework (SSDCF) based on AI approach and Typeface Analysis

Mrouj Almuhajri

A Thesis

in

The Department

of

Computer Science and Software Engineering

Center of Pattern Recognition and Machine Intelligence CENPARMI

Presented in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy (Computer Science and Software Engineering) at

Concordia University

Montréal, Québec, Canada

September 2022

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Mrs. Mrouj Almuhajri**

Entitled: **Shop Signboards Detection and Classification Framework (SSDCF) based on AI approach and Typeface Analysis**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Computer Science and Software Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

____________________ Chair
Dr. John Zhang

____________________ External Examiner
Dr. Eric Granger

____________________ Examiner
Dr. Mingyuan Chen

____________________ Examiner
Dr. Adam Krzyzak

____________________ Examiner
Dr. Tse-Hsun (Peter) Chen

____________________ Supervisor
Dr. Ching Y. Suen

Approved by ____________________
Lata Narayanan, Chair
Department of Computer Science and Software Engineering
Center of Pattern Recognition and Machine Intelligence CENPARMI

September 14, 2022

____
Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Shop Signboards Detection and Classification Framework (SSDCF) based on AI approach and Typeface Analysis

Mrouj Almuhajri, Ph.D.

Concordia University, 2022

Rapid advancements in artificial intelligence algorithms have sharpened the focus on street signs due to their prevalence. This research was driven by beneficial applications of end-to-end systems to humans, municipal agencies, and automobiles. However, the variation of materials, shapes, colors, and fonts in some signs, such as shop signboards, have presented complicated challenges to AI-based systems to detect and classify them. Previous studies built classification models by considering the whole storefront. Their classification results were negatively impacted by the inclusion of other components within the storefront. This research focuses on shop signboards as they are much more consistent.

The main objective of this research is to detect and classify shop signboards based on deep learning and machine learning techniques. To achieve that, data acquisition was necessary for models training purposes. Therefore, the Shop Signboard ShoS dataset was collected from Google street images. A total of 10k store signboards were captured within 7500 images. All the collected images were fully annotated and made available for the public for several research purposes.

Then, the Shop Signboard Detection and Classification Framework SSDCF was designed and built to tackle most of the existing challenges. Three main components were fully implemented and evaluated: signboard detector, text extractor, and shop classifier to classify commercial stores based on the textual information. For signboard detector, two models were trained and tested utilizing the ShoS dataset. Findings surpassed the performance of YOLOv3 without any color preparation.

For text extractor, the evaluation of Google Vision OCR showed better results even with the existence of influential factors, such as stylized fonts and skewed images. For shop classifier, out of the two trained and tested classifiers, SVM showed great performance even with classes that have some difficulty factors. The performance of the classifier had been enhanced by 4% approximately after adding the augmented data which was generated by the Random Deletion method and a novel Thesauruses-inspired method named *OCR-Thesauruses*. Each component has been trained and tested individually at first. Then, the full end-to-end framework was implemented and evaluated using the SVT public dataset, and the outcome reached an F1-score=89%. The classification performance was also compared with human performance based on the texts extracted from the signs. Human subjects were provided with textual information only and were not exposed to shop sing images. The results showed that our classifier exceeded human performance by about 15% due to the prior knowledge the classifier learned from all text data during training.

Finally, the results of the second component of our framework, the text extractor, were statistically analyzed to check the impact of typeface styles used in shop signboards on the recognition rates. The findings showed a significant association between the typeface style and the recognition rate. So, it is recommended to use ”Serif” and ”Sanserif” styles over ”Script” and ”Decorative” in designing shop signboards. If using stylized fonts is a must, it is advised to add keywords that distinguish a store class from another using a better typeface design, such as ”Serif” or ”Sanserif” styles.

Acknowledgments

The journey of my Ph.D. went through many ups and downs. Reaching the end of this journey obliges me to show my gratitude to many people and parties. First and foremost, I am extremely grateful to my supervisor, Prof Ching Y. Suen, for his unlimited support, valuable advice, and patience during my Ph.D. study. During my academic research trip, he was always there providing his immense knowledge and plentiful experience. I would also like to thank CENPARMI's lab manager Nicola Nobile for always being there for any technical support. I also appreciate all the support I received from the CENPARMI members as they are my family here abroad, especially Afnan, Najla, Rabia, and Hiba who shared the same labs with me during this long journey. I thank them for the precious time we spent together discussing research and life matters.

My gratitude extends to my lovely husband Mohannad Bamoallem who stood by my side in each and every step even when he was far away. His support and love showered me all the time, without him I would not be here. I also would like to give a big thank to my daughter Jood who started this journey with me since her first day of her life. She brought joy through tough times and motivated me to keep working and success. I also would like to thank my parents and siblings for their unconditional love and support. I would not be able to reach this point without their support.

Special thanks to my friends Alaa, Hanan, Reem, Sabreen, and Somyah for their encouragement and support. I am so grateful for the precious value they added to my life by listening, discussing, and supporting. Thanks are never enough to show my gratitude to them.

Finally, many thanks go to, Saudi Electronic University and NSERC (Natural Sciences and Engineering Research Council of Canada) for their financial support. Also, my appreciation also goes to Saudi Cultural Bureau for facilitating my needed processing during my stay in Canada.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As technology advances with the enhancement of artificial intelligence algorithms, many processes can be automated. This increase in automation has improved the quality of our life by enabling efficiency and time saving in our personal and professional lives. Nowadays, complex algorithms can be executed in real time on contemporary mobile devices to process our surrounding environment and return with additional intelligent information. For example, the i-street application [14] was introduced to detect and distinguish street signs in a video stream, and then used augmentation methodology to show some points of interest in real time utilizing location service. Also, self-driving vehicles can leverage deep learning techniques to read and analyze traffic signs and perform the needed adjustments based on them.

Street signs can be categorized as controlling, naming, informing, selling, servicing, or commemorating signs [15]. Some signs have consistent shapes and pre-defined colors and fonts, such as traffic signs which are categorized as controlling signs. This consistency makes it easier for systems to adopt machine learning and deep learning techniques to detect and identify them [16, 17, 18]. Other signs may have variation of materials, shapes, colors, and fonts like naming signs (i.e. shop signboards). This variation presents a more complicated challenge for AI-based systems to detect and identify those signs.

Naming signs including store signboards represent about 41.5% of other street sign categories

[15]. Shop signboards are the key in identifying services and goods provided by those shops. Street view imagery has quickly risen to prominence as a valuable data source for training deep nets and making informed decisions. The abundance of street views has already spurred many automated systems that use commercial listings to provide consumer-driven recommendations and ratings such as Yelp[1]. However, the rapid change in the market is causing shops to open and close at a much faster pace now. This is very evident when examining the food business where about 30% of restaurants are closing on an annual basis [19]. A recent study showed that 43% of small businesses were closed in the US during the COVID-19 pandemic [20]. This rapid change presents a unique challenge where it is important to have an automated mechanism for identifying and classifying shops that are not dependent on commercial listings.

This chapter states the thesis problem in Section 1.1, discusses the motivation and application in Section 1.2, indicates the objectives and contributions in Section 1.3, and lists assumptions and limitations in Section 1.4. At the end, the overall thesis outline is given.

1.1 Problem Statement

Using machines to detect and classify stores based on their signboards presents a complex problem due to the large number of factors that can influence the process [12, 13]. For example, shops' signboards are known to take various shapes and forms regardless of the guidelines to best design signboards in order to improve their uniqueness, readability, and attractiveness for potential customers [21]. This visual variability makes it challenging for machines to differentiate between stores of different types. In addition, storefronts may have irrelevant information that can mislead the learning process for machines. For instance, a clothing store may have an ATM in its storefront and that could confuse the model to classify the store as a Bank as was the case in [12]. Furthermore, although there is an abundance of street views that contain storefronts, the task of collecting and annotating them by humans is hard, time consuming, and prone to errors because of the intensive work and resources it requires [11, 12, 13].

[1] https://www.yelp.com

Typeface designs for road traffic signs and license plates have been the subject of multiple studies and recommendations [22, 23]. However, typeface designs for shops' signboards have not gotten much attention as they usually are aligned with guidelines for human readability and overall layout. Businesses around the world are emphasizing the utilization of Artificial Intelligence (AI) technology in several life-related aspects. This direction is putting more emphasis on typeface designs used in store signboards and their compatibility with machine-based systems.

Based on the challenges and gaps mentioned above, this research proposes a system that detects and classifies business store signboards based on existing deep learning and machine learning algorithms. The framework consists of three main components: 1) signboard detection, 2) text extraction, and 3) store classification. In addition, this research statistically investigates the impact of signboards' typeface designs on their recognition.

1.2 Motivation and Application

AI-based systems enable reading text from street views efficiently [24] which allows various applications that can be used by commercial organizations or government agencies for several purposes. This research was driven by possible beneficial applications of these systems for humans and municipal agencies in addition to automobiles.

Being able to detect shop signboards can allow us to classify them and identify areas of interest within them. Different types of users would benefit from these applications, such as tourists and visually impaired individuals. According to the World Health Organization[2], there are 2.2 billion people at least around the world living with vision impairment or blindness. Such AI-based systems can improve the quality of their lives by leveraging smart phones to explore new neighborhoods and recognize any store. This can be done simply by taking a picture of storefronts regardless of how they see it or in which language the signboard is without the need for human assistance.

[2] https://www.who.int/en/news-room/fact-sheets/detail/blindness-and-visual-impairment

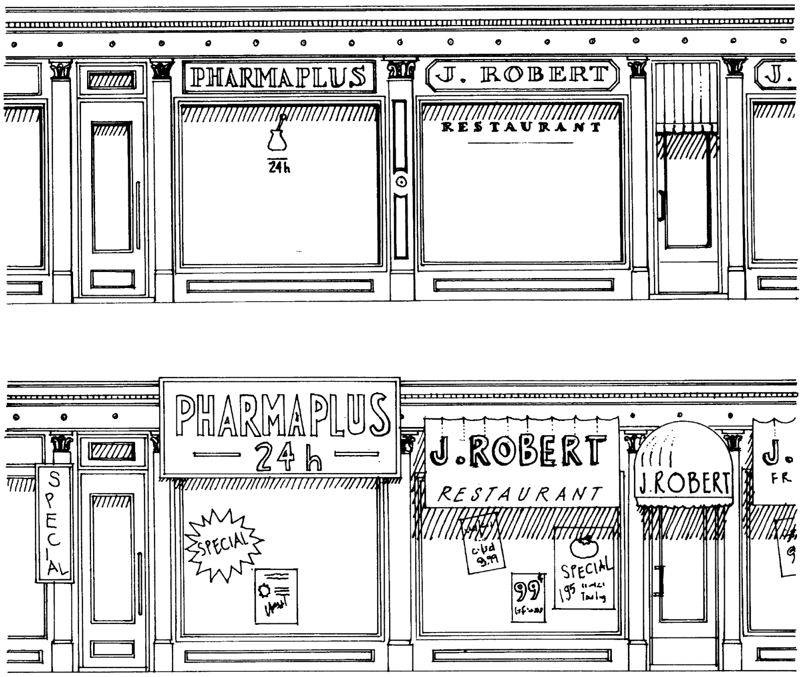


Figure 1.1: A sample of well integrated shop signs (top), and the jumbled shop signs (bottom) provided by the City of Westmount[1]

Municipal government agencies issue policies and regulations to govern the design of storefronts and their signboards. These policies and regulations are usually different from one neighborhood to another. Human inspectors are usually utilized to inspect adherence to such regulations; however, this inspection mechanism can be time consuming and prone to human errors. Hence, a system for identifying and classifying shop signboards can improve the speed and accuracy of this process. For instance, the City of Westmount in Quebec issued regulations governing the design of storefronts and their signboards and mandating their harmony with the architectural design in some specified streets [1]. These guidelines include language, size, lettering, and graphic elements. Figure 1.1 illustrates the acceptable/unacceptable shop signage by the law of the City of Westmount.

Furthermore, despite the fact that self-driving vehicles are already utilizing AI-based systems in many aspects of their operation including the identification of points of interest based on location, there are still some other areas of their operation that can benefit from such systems. For example, some local stores are not registered on maps, or their information is not updated due to the rapid change in business openings and closings as highlighted before. Therefore, self-driving cars that are equipped with front, backward-facing, and side-facing cameras may utilize the proposed systems to provide a more comprehensive analysis of their surroundings.

Text in natural images is the core clue for the store classification process as it contains rich semantics that are frequently highly relevant to shop class. Although recognizing scene text has been an ongoing research topic for decades, many studies focused on the consistent appearance of text like in traffic signs [25] and license plates [23]. However, other signs like shops signboards have gotten less attention. The complex visual appearance of shop signboards needs to be investigated more thoroughly to enhance several applications. In [12], the trained model considered the whole storefront for classification purpose; but the results were negatively impacted due to the inclusion of other components within the storefront. Hence, this work focused only on the text of store signboards as it is much more consistent than the whole storefront.

1.3 Objective and Contribution

Despite the large number of street view images provided by Google, collecting storefront images and annotating them is a hard and time consuming task that is prone to human errors due to the intensive work it requires. Literature shows that detecting and classifying storefronts from Google street view images is often negatively impacted by the lack of annotations or having inaccurate ones [11, 12, 13]. This research aims to collect a considerable amount of required data with accurate and full annotation for training and testing the proposed system. The ShoS dataset, which is fully described in Chapter 4, will be made available to public for research purposes including but not limited to shops' signboard detection, store classification, text recognition, and typeface classification.

Moreover, the main objective of this research is to detect and classify store signboards based on deep learning and machine learning techniques. Unlike traffic sign classification problems [25], there is no standard way of representing store signboards. Some of the shop signboards follow the surrounding environment especially for urban cities [26], and some stick to the traditional sign with style and writing close to inscriptions [15]. Different shapes, materials, colors, and fonts make it challenging to detect signboards and extract useful information out of them for store classification purposes. Our Shop Signboard Detection and Classification Framework SSDCF was designed and built to tackle most of these challenges by pipeline models based on machine and deep learning algorithms in order to detect shop signboards, extract text out of them, and classify stores. The classification results were enhanced using a novel methodology Details about our methodology are provided in Chapter 5.

Furthermore, most of the studies done on store signboards and their typefaces have focused on the human side only and were bound by the application of psychological theory [27], such as signage readability and attractiveness to customers. This research examines the machine side by statistically investigating the impact of typefaces used in store signboards on the ability of systems to recognize them. This investigation would help future business owners in adjusting their storefront designs in order to capitalize on the proliferation of AI-based applications and systems.

1.4 Assumption and Limitation

For data collection:

- All signboards are rectangular and located on top of their storefront.
- The research scope is limited to six categories of shops only (illustrated in Figure 4.5).
- Only English text was recorded in the ShoS dataset.
- Numbers and addresses appeared in shop signboards were not recorded in the ShoS dataset.

For deep learning techniques:

- Machines with a graphical processing unit GPU and high RAM memory are required in order to train the designated models in a reasonable time. Although we managed to utilize a machine with adequate GPU and RAM, the size of input images was reduced in order to enable the processing of the models. The reduction in image sizes might impact the performance of the models.

For typeface analysis:

- Although six type styles (i.e. classes) were considered in this research, two styles were eliminated "Blackletter" due to the lack of samples, and "Mix" style in the final analysis because of inconsistency in the number and type of styles used in these classes.
- To avoid the influence of any external factors in the analysis phase of typefaces, all samples that are "occluded" or defined as "difficult" were list-wise eliminated.

1.5 Thesis Outline

In this thesis, we implemented a full methodology to detect and classify stores based on their visual signboards named *Shop Signboard Detection and Classification Framework SSDCF*. In addition, a detailed analysis was conducted on the used typefaces to investigate the influence of type styles on the recognition process. Chapters will be organized as follows. First, Chapter 1 states the thesis problem, discusses motivation and application, indicates objectives and contributions, and lists assumptions and limitations. Then, Chapter 2 gives a brief background about object detection, text classification, and typeface design for non-specialized readers . Next, Chapter 3 includes in extensive detail the literature work done in the research fields. After that, Chapter 4 elaborates on the phases of collection and annotation of the ShoS dataset. Then, Chapter 5 explains the framework and methodology of this research followed by Chapter 6 which illustrates and discusses the results. Moreover, Chapter 7 provides an analytical study about the influence of typeface design on the recognition rate.Finally, Chapter 8 concludes this work and provides future insights.

Chapter 2

Background

This research combines different fields together including object detection and classification, text extraction, natural language processing, and typeface design. This chapter gives a brief overview about the mentioned fields for non-specialist readers. Section 2.1 elaborates essential concepts and terminology in object detection. Then, Section 2.2 details methodology of text extraction and classification. Finally, Section 2.3 presents comprehensive definitions about typeface design.

2.1 Object Detection

In computer vision, object detection can be defined as the task of analyzing digital images and detecting instances of visual objects of interest belonging to certain classes. It simply answers two questions about the object: 1) what is it? (detection) and 2) where is it? (localization). Although the field of object detection has been investigated for decades, it gets a great boost recently after deep learning techniques are involved. Object detection is simply trying to draw a bounding box around the target object in order to locate it within the image and classify it. The difference between object detection and classification is that the first one can detect (locate and classify) several objects within the image (the number of objects can vary per image) while the latter classifies a single class, see Figure 2.1 for more clarification. Object detection has found applications across several fields, such as face detection, pedestrian detection, traffic sign detection, and even text detection. Figure 2.2 shows some examples of object detection applications.

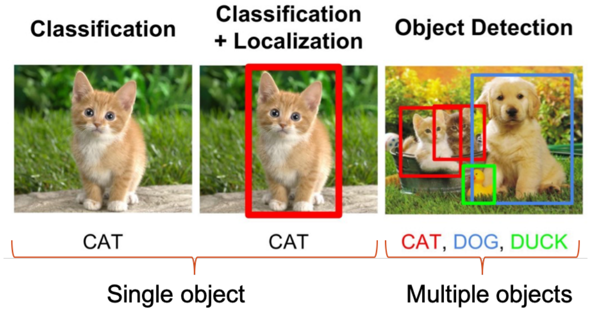


Figure 2.1: The difference between classification and object detection [2]

Object detection started almost 20 years ago with traditional detectors in which the process of detection was based on handcrafted features to represent the image and hence process it. Due to the limitations of computer resources at that time, traditional detectors were not as efficient as they were supposed to be. They should have to make a trade-off between performance and speed. Recently, the new generation of object detectors is based on deep learning techniques (i.e. Convolutional Neural Network CNN). With the availability of good resources, detectors can reach real-time detection with good performance. The history of CNN-based detectors including the RCNN family, YOLO, and SSD is discussed intensively in section 2.1.1.

In this research, the focus will be on the one-stage detectors, so a brief introduction about its main components is given as follows. In one-stage object detectors, a given input image is fed into a model to make several predictions in a single pass with some big convolutional network. The

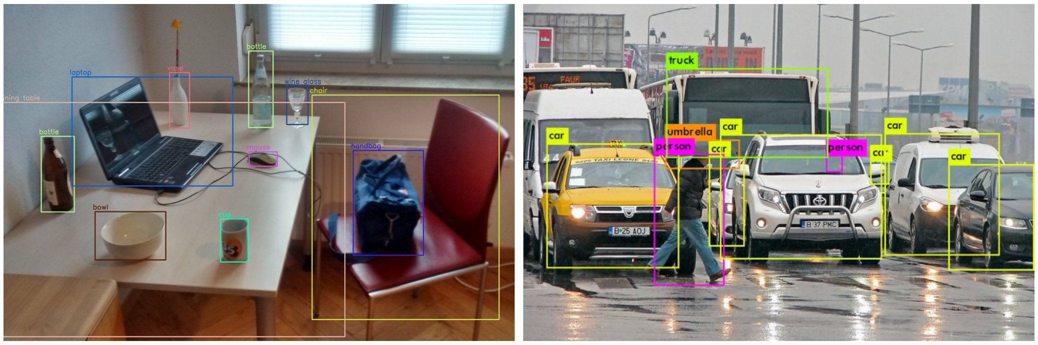


Figure 2.2: Examples of object detection applications

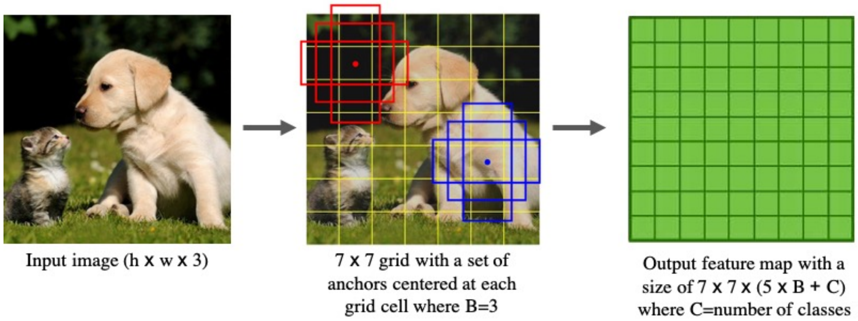

Input image (h x w x 3)
7 x 7 grid with a set of anchors centered at each grid cell where B=3
Output feature map with a size of 7 x 7 x (5 x B + C) where C=number of classes

Figure 2.3: An abstract view of one-stage object detection concept

input image is divided into one or more grids (e.g. 7×7), and a set of base boxes (i.e. red and blue boxes in Figure 2.3 are defined for each grid cell). These base boxes are known as '**anchors**' or '**default boxes**', and their center is the same as a cell center. Also, the number of these anchors B (e.g three) is chosen before training and can not be changed later on during training. Likewise, the size of anchors varies (i.e. tall, wide, square), and it is fixed. Usually, anchors are nothing but widths and heights, and they describe the most common object shapes in the dataset.

In practice, there are two common ways to describe the coordinates of bounding boxes with four numbers: 1) $(xmin, ymin)$ which represents the top left corner, and $(xmax, ymax)$ which represents the bottom right corner, and 2) $(center_x, center_y)$, width, and height. Using the grids and anchors, the model is able to make several predictions for bounding boxes with offsets, confidence score, and class probabilities for each box. The number of predictions is the grid size multiplied by the number of detectors (i.e. anchors). For example, with a grid of 7×7 and 3 detectors, it gives us 147 predictions.

The **offsets** predictions for bounding box coordinates are to show how larger/smaller the ground truth box is than the anchor. In addition, the **confidence score** prediction (AKA objectness score) reflects how likely the model thinks the predicted box contains an object, and it is a number between 0 and 100 (the higher the better). The confidence score can be computed based on **intersection over union IoU** (AKA Jaccard index) to see how the predicted box matches the ground truth. Finally, a **class probability** distribution for each class of interest in the dataset is calculated using some activation functions like softmax. Therefore, a giant tensor of size $7 \times 7 \times (5 \times B + C)$ is produced, where B is the number of anchors, C is the number of classes in the dataset, and 5 represents the four coordinates for the bounding box plus a confidence score.

2.1.1 Object Detectors Timeline

In the object detection field, many algorithms have been developed to find the occurrences of objects within an image or a stream in the fastest and most efficient way. Over time, the object detection field has been through two main stages: traditional and deep learning. The breaking year between the two phases is 2014. Regardless of the hard work that was done in the first stage, there is significant acceleration in the development during the second stage. Approaches are categorized as two-stage and one-stage detectors. Below, they will be discussed in detail.

Two-stage Detectors

A convolutional neural network CNN is one of the most common methods in deep learning. It learns robust and high-level feature representations of an input image. As mentioned above, we can not simply use CNN in object detection because the number of occurrences of an object is not consistent plus it may have different spatial locations and different aspect ratios. Girshick *et al.* were pioneers in bringing CNN to object detection by proposing Regions CNN or what is called RCNN [28, 3]. RCNN and the next updated versions perform object detection through a pipeline of multi-step series: 1) finding region proposals; 2) verifying object positions and classifying objects within those regions. Figure 2.4 shows the differences in the architecture of the RCNN family.

- **RCNN Regions Convolutional Neural Network**

  RCNN [28, 3] can be simply described as follows: a selective search is used in order to extract region proposals (2k only). Then, these region proposals are re-scaled to a fixed size and fed into a CNN already trained on the ImageNet dataset. Therefore, the CNN model produces a long multi-dimensional feature vector as output after fine-tuning using log loss. Finally, the extracted features are fed into several linear binary Support Vector Machines SVMs for each

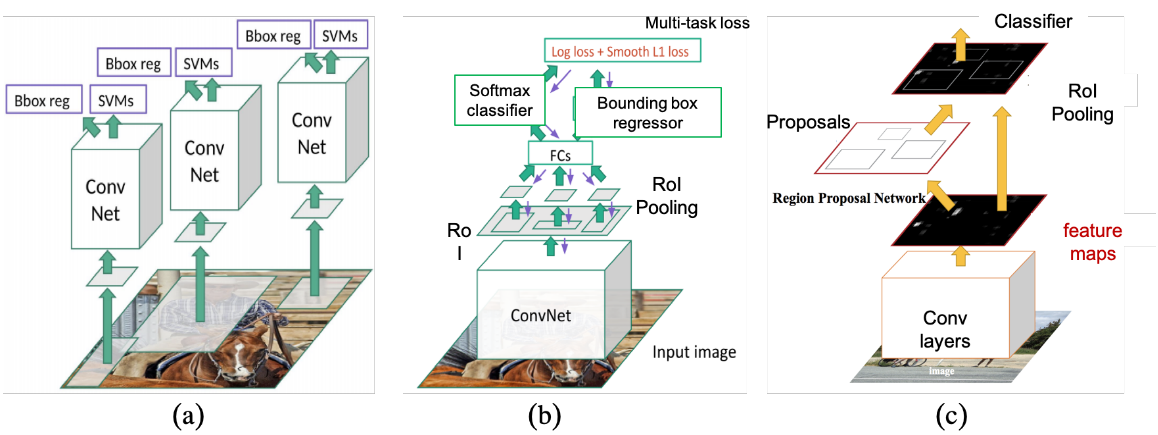


Figure 2.4: The abstract architecture of RCNN family (a) RCNN [3] (b) Fast RCNN [4] (c) Faster RCNN [5]

class to predict the existence of an object within that region proposal and to recognize it. In addition to that, there is also a bounding box regressor in order to add a corrective feature to the bounding boxes.

Although RCNN has dramatically improved the mean average precision (mAP=62.4%) by more than 50% compared to the best results on the PASCAL VOC2012 dataset [3], one major drawback about this approach is its extremely slow detection speed. This is because of the redundant feature computations for the overlapped region proposals.

- **SPPNet Spatial Pyramid Pooling Networks**

  SPPNet came to solve RCNN problem. It was proposed by [29] to overcome the fixed input size required by the CNN. Regardless of the input image size/scale, SSPNet works similar to RCNN except that the feature maps are extracted from the entire image only once. Then, a spatial pyramid pooling is applied on each candidate window of the feature map to get a fixed length representation of that window. By doing this, the redundant computations are avoided and hence the speed increased by 20 times compared to RCNN. However, the average mean precision mAP is still very close to RCNN. Also, the fine-tuning can be done in the fully connected layers only which limits the accuracy.

- **Fast RCNN**

  Fast RCNN [4] is the improved version of RCNN [3] where it takes into count the refinement done in SSPNet [29]. The main contribution of Fast RCNN is that it enables the detector and the bounding box regressor to work simultaneously under the same network configurations in training. Therefore, training can be done in a single stage with multi-task loss (log loss for the detector + smooth L1 loss for the regressor). In addition, fine-tuning can be done through all network layers.

  Results of training the VGG16 with Fast RCNN are three times faster for training and 10 times faster for testing compared to SPPNet. It also gives more accurate results than SSPNet. Nevertheless, detection speed is still limited because of the region proposal step which consumes almost the same time spent on the detection network considering that this step has not been done with a CNN model up to this point.

- **Faster RCNN**

  All the previous algorithms are using selective search to find out the region proposals which slows down the algorithm. So, the idea of Faster RCNN came by Ren *et al.* [5] as they proposed an algorithm eradicates selective search and uses a deep net instead. It is called Region Proposal Networks RPN. The input image is fed into a convolutional net to get the feature map which in turn is fed into RPN to generate region proposals based on anchor concept (i.e. boxes with different sizes and different aspect ratios). Then, the candidate region proposals are reshaped using the RoI pooling layer which is used next for both image classification within the proposed region and bounding box regression.

  Faster RCNN is the first end-to-end deep learning detector, and it has been tested on PASCAL VOC 2007 and 2012. It achieved high performance in terms of speed and accuracy reaching mAP= 73.2% and 70.4% for the mentioned datasets respectively. Despite this successful enhancement in object detection, redundancy in computations is still occurring at the subsequent detection stage [30].

One-stage Detectors

Unlike the previous detectors, all algorithms in this group are implemented in one stage. In particular, they are based on regression in which the whole image is scanned and predictions are made to localize, identify, and classify objects simultaneously. Hence, real-time object detection can be achieved with a single neural network.

- **YOLO You Only Look Once**

  YOLO was proposed by Redmon *et al.* [31], and as its name says, the idea behind it is to look at the input image only once! YOLO divides the input image into an $S \times S$ grid (the default 7x7), so each cell in that grid is responsible for predicting B bounding boxes (default 2). To explain, if we have a grid of 13x13 with 169 cells and each cell generates 5 bounding boxes, then there will be 845 bounding boxes that are made at the same time. Each bounding box is coupled with its confidence score. A higher confidence score indicates a high possibility of an object existing within this bounding box. It is calculated using formula (1).

  $$Confidence\ Score = P(Object) \times IoU_{truth\,pred.} \tag{1}$$

  For each bounding box, the cell also predicts a C conditional class probability given that the grid cell contains an object using the pre-trained CNN classifier. That is, the features are extracted from the input image in the initial layers and the fully connected layers give the probability distribution over all other possible classes $P(Class_i|Object)$ in addition to the coordinates. To get class-specific confidence score for each box at the test time, formula (2) is used.

  $$P(Class_i|Object) \times P(Object) \times IoU_{truth\,pred.} = P(Class_i) \times IoU_{truth\,pred.} \tag{2}$$

The model was evaluated on PASCAL VOC detection dataset [32] with 24 convolutional layers and 2 fully connected layers in a framework called Darknet. Another format of the model called Fast YOLO was also proposed and evaluated on the same dataset with fewer convolutional layers, 9 in particular, and fewer filters for even faster object detection. Results show great enhancement in speed reaching real-time in which YOLO processes 45 fps and Fast YOLO runs at 155 fps. They could also reach 63.4% and 52.7% mAP for YOLO and Fast YOLO respectively.

YOLO as the first detector in the one-stage group is extremely fast compared to the two-stage detectors. Considering the fact that YOLO sees the whole image makes it less vulnerable to background errors. Thus, it can generalize to new domains or unexpected inputs. Yet, it suffers from localization accuracy drop compared to Fast RCNN, especially for small objects. A series of updated versions have been proposed later on introducing YOLOv2 (AKA YOLO9000) [33], YOLOv3 [6], and YOLOv4 [34] in order to improve the limitations.

In **YOLOv2** [33], the same concept of anchor boxes, that is used in Faster RCNN, is used with some modifications. In particular, k-means clustering has been used to find anchors rather than hand-picked. In addition, the grid becomes smaller as of 13x13 and that leads to more fine grained features which enhance the detection of small objects. The number of predicted bounding boxes per cell is set to five instead of two in YOLOv1. Furthermore, the input size has increased to 448x488 instead of switching between 244x244 and 488x488 for dense layers in YOLOv1. So, less computations have been achieved and hence higher speed. Moreover, the number of convolutional layers has decreased to 19 using Darknet-19. Also, multi-scale training has been considered in order to improve detection accuracy for the same objects with different scales. At that point, YOLOv2 defeated Faster RCNN and SSD in terms of speed and mAP reaching 76.8 mAP on VOC2007 at 67 fps and 78.6 mAP at 40 fps.

In **YOLOv3** [6], anchor boxes are still used with logistic regression to predict bounding boxes

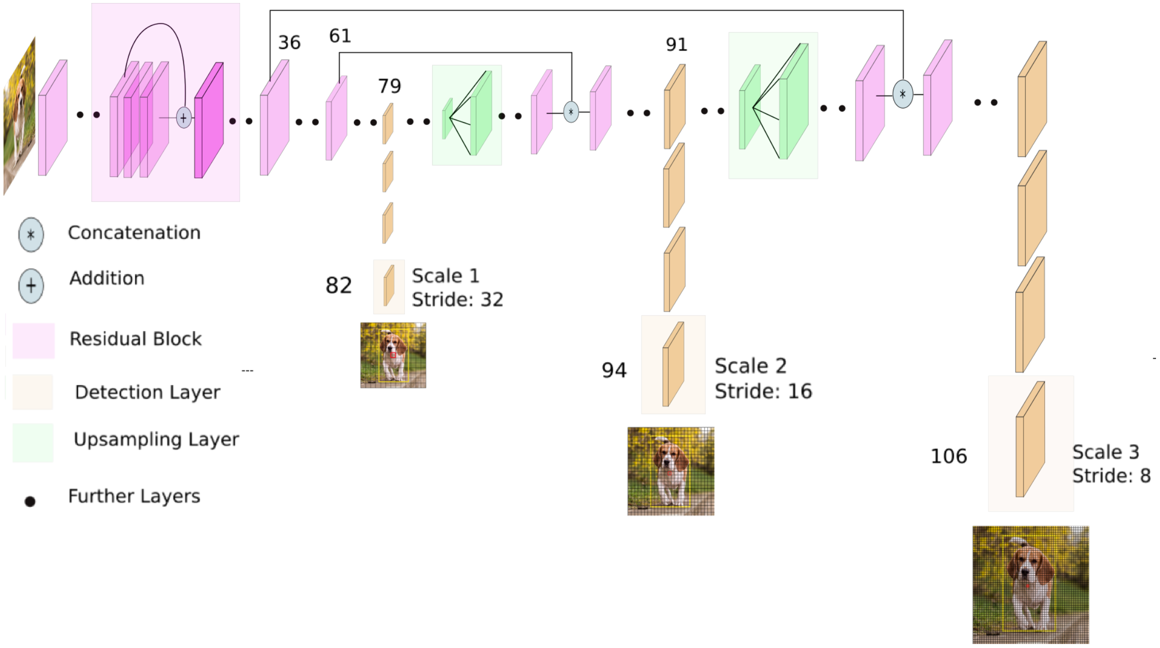


Figure 2.5: The architecture of YOLOv3 [6] retrieved from [7]

at different scales. The number of predicted bounding boxes per cell is set to three instead of five in YOLOv2 for the sake of higher speed. The detection is done on feature maps of three different scales which is one of the most salient features of YOLOv3 hence the tensor is $N \times N \times [3 \times (5 + C)]$, in which N represents the grid size, 3 is the number of predicted bounding boxes (anchors), 5 represents the four bounding box coordinates plus the object confidence score, and C is the number of classes. To increase the accuracy of class prediction, independent logistic classifiers for each class have been used instead of softmax which enables multi-label classifying. The loss function has changed also from v2 to v3 as the former uses the squared errors while the latter uses cross-entropy errors. That means object confidence score and class predictions are predicted through logistic regression. The architecture of v3 is more powerful with 53 convolutional layers using Darknet-53 which leads to better accuracy. Comparing YOLOv3 to RetinaNet, it has a similar performance of 57.9 AP_{50} on a Titan X but 3.8x faster. Yet it is important to consider that Darknet-53 could consume more time than Darknet-19 in v2 regardless the better accuracy. Figure 2.5 illustrates the architecture of YOLOv3 which we used in our research.

In **YOLOv4** [34], a novel backbone CSPDarknet53 is used to enhance the capability of CNN learning to detect multiple objects of different sizes. On top of the CSPDarknet53, a spatial pyramid pooling block is added for feature extraction in order to broaden the receptive field and isolate the most leading context features as it assists in covering the increased input size. For the neck component, the PANet path-aggregation is used for different detector levels instead of feature pyramid networks (FPN) used in YOLOv3 to increase accuracy. The head, which is used for locating bounding boxes and classifying them, is the same one used in YOLOv3. By testing YOLOv4 on Imagenet [35] dataset for classification and on MS COCO [36] dataset for detection. They reported an improvement in mAP under certain parameters and GPUs by as much as 10% for MS COCO. Also, the number of frames per second was enhanced by 12% approximately.

- **SSD Single Shot MultiBox Detector**

  SSD was initially introduced by Liu *et al.* [9] as one-stage deep learning detector. Its major contribution to object detection is the concept of multi-references and multi-resolutions. It can be simply explained as pre-defining a set of default boxes (i.e. anchor boxes) with different sizes and aspect ratios at several locations of the image. Then, bounding boxes are predicted based on these references. For extracting features, SSD uses VGG-16 model which has been pre-trained on ImageNet as a base. Then, several convolutional feature layers are added which downsamples the image, unlike YOLO which upsamples it. This is known as a pyramid representation of images at different scales. Figure 2.7 shows more details about SSD architectures. SSD predicts bounding boxes using several grids (the number and size of grids may vary depending on the exact model architecture) with different scales. For example, MobileNet-SSD model architecture has six grids: 19×19, 10×10, 5×5, 3×3, 2×2, 1×1 where the largest grid with smaller cells at earlier layer is responsible for extracting fine-grained feature maps which are good for small objects. In contrast, the smallest grid which takes the entire image at the last layer is responsible for extracting coarse-grained feature maps to detect large objects. Figure 2.7 gives an example from the original paper of SSD [9]

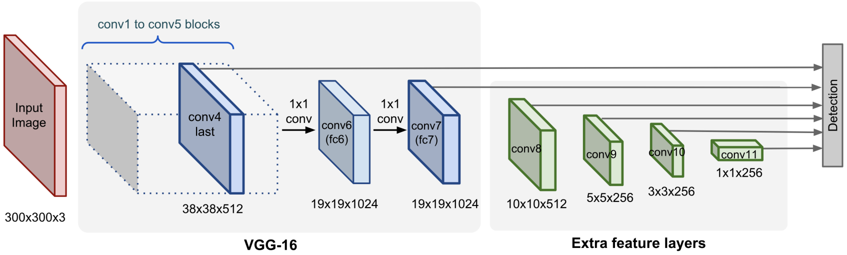
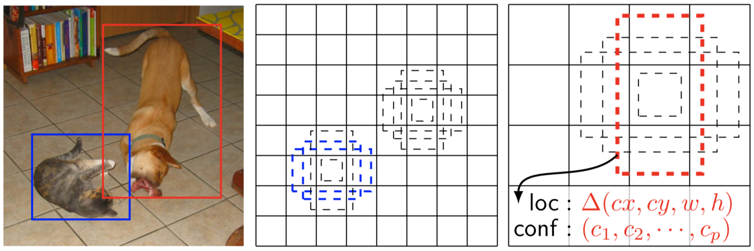


Figure 2.6: The architecture of SSD retrieved from [8]


loc : $\Delta(cx, cy, w, h)$
conf : $(c_1, c_2, \cdots, c_p)$

(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

Figure 2.7: An example from the original paper of SSD [9]: (a) Ground truth bounding boxes. (b) Anchor boxes in a fine-grained map to detect smaller objects (cat). (c) Anchor boxes in a coarse-grained map to detect larger objects (dog).

in which the dog can be detected in the smaller grid (4×4), and the cat in the lager grid (8×8).

In comparison to YOLOv3, which has three grids and makes three predictions per cell, SSD makes 3-4 predictions per cell for the larger girds and 6 predictions per smaller grids. the number of predictions in SSD is much more than YOLO due to the increase in the number of grids. For instance, SSD512 model outputs 24,564 predictions in comparison to 845 in YOLOv3. Although that would lead to better accuracy, the post-processing may affect the speed. Moreover, while YOLOv3 predicts 4 bounding box offsets, one confidence score, and class probabilities as mentioned before, SSD does not predict the confidence score. Instead, it has a special background class that represents a low confidence score in YOLO. Hence, the loss function is quite different in SSD as there is no confidence score. The overall loss function in SSD is the sum of localization loss and classification loss, see formula (3):

$$L = \frac{1}{N}(L_{class} + \alpha L_{loc}) \tag{3}$$

where N is the number of matched bounding boxes; localization loss is computed by Smooth L1 loss; α is the weight term to balance between the two losses and it is picked by cross validation. In regard to classification loss, it is softmax over multiple classes.

After testing SSD models using several datasets, such as PASCAL VOC and COCO, results reveal that SSD512 model excels Faster RCNN in term of accuracy (76.9% mAP) and speed (3x faster). However, the later versions of YOLO are still competing in speed.

- **RetinaNet**

  Between two-stage and one-stage detectors, there is some trade-off between accuracy and speed. Despite the success of one-stage detectors like YOLO and SSD with real-time speed, the accuracy they achieved is 10-40% relative to the state-of-art of two-stage detectors[37]. Hence, RetinaNet [37] came up with a novel idea to improve the accuracy. Actually, the most likely cause of that issue is the imbalance between foreground and background classes during training. In the two-stage detectors, the proposal stage is responsible to narrow down the number of candidate proposals to 1-2k approximately in contrast to one-stage detectors which work differently and may come up with around 100k proposals. This would affect the one-stage detectors from competing with the state-of-art of the two-stage detector in terms of accuracy. RetinaNet proposed a novel loss function called Focal Loss which takes in its consideration the mentioned facts. Focal loss reshapes the standard cross-entropy loss so that the detector will put more weight on hard examples (i.e. background with noisy texture or partial object) and down-weight easy examples (i.e. empty background) during training. By using focal loss, Retina could achieve comparable accuracy and speed when it was evaluated using COCO dataset and compared to two-stage and one-stage detectors.

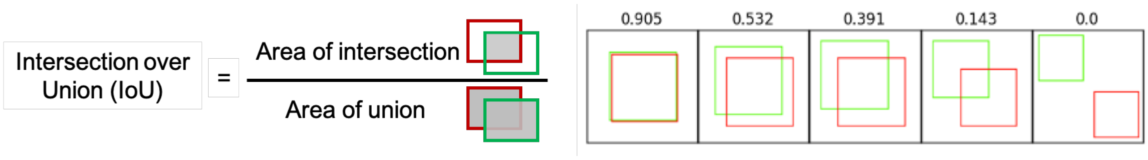


Figure 2.8: Intersection over Union IoU where green boxes represent the ground truth and red boxes is the predicted bounding boxes with some scenario samples of IoU values

2.1.2 Evaluation Metrics

The way object detection models are evaluated is quite different than classification and regression problems as we have more complicated tasks (i.e. localization and classification). Therefore, we need to evaluate both localization (predicted bounding boxes) and classification (the presence of an object and its class). First, localization is measured based on **Intersection over Union IoU** which is the area of intersection for the predicted bounding box and its ground truth divided by their area of union (see Figure 2.8). The prediction is considered correct based on a predefined threshold (usually 0.5 or higher). Thus, if the IoU is greater than that threshold, the prediction is considered correct (**True Positive TP**). Otherwise, they are **False Positive FP**. If the model failed to detect an object that is present in the ground truth, it is called **False Negative FN**. Any remaining part of the image that the model did not predict is called **True Negative TN** which is not a concern in object detection.

So, using the information of TP, FP, and FN, we can calculate Precision and Recall as it is shown in the following equations 4 and 5. In general, **precision** gives the proportion of ’if the predicted positives are truly positive?’, and **recall** provides the proportion of “if the actual positives are correctly classified?”

$$Precision = \frac{TruePositive\ TP}{TruePositive\ TP + FalsePositive\ FP} \tag{4}$$

$$Recall = \frac{TruePositive\ TP}{TruePositive\ TP + FalseNegative\ FN} \tag{5}$$

Using the previous information, it is possible to compute **Average Precision AP** over several thresholds. AP is originally introduced in the PASCAL VOC2007 challenge [38], and it is defined as the average precision of detection taken under different recalls. In other words, it is the area under the precision-recall PR curve. AP can be calculated for each class, or overall classes as the **mean average precision mAP**. The final performance is computed using mAP based on IoU with one predefined threshold. Yet, in MS-COCO dataset [39], mAP is computed over multiple IoU thresholds ranging between 0.5 for coarse localization and 0.95 for perfect localization.

In addition, the speed of the object detection algorithm is considered when evaluating models, especially with the enhancement of real-time systems. The work in [40] shows that gaining more accuracy for the compared object detection models includes Faster RCNN [5] and SSD [9] means sacrificing the speed. So, up to date, evaluating the running time performance depends on many factors like the feature extraction method, and the hard gear used when testing the system.

Finally, some studies like [13, 41, 12] compared their work with human accuracy in addition to the above metrics in order to evaluate their proposed methods. The model is well-performed if it is able to achieve comparable human-level accuracy.

2.2 Text Extraction and Classification

2.2.1 Optical Character Recognition

Text is the core of shop signboards and sometimes it is possible to convey rich and beneficial information on what products a shop is selling or which services it provides through text only. Optical Character Recognition OCR was introduced to extract text from digital images, which are acquired by digital input devices like cameras or scanners, and transform it into machine-readable text. That would allow more useful tasks on the extracted text like searching and classifying them. Scene text recognition STR is a sub-field of OCR, and it has been widely addressed over the last few decades yet is still challenging because of the complexity coupled with it [42]. Different backgrounds, several fonts, and some environmental noise can affect OCR performance. Fortunately, with the advance in deep learning techniques in the computer vision field, OCR algorithms have been developed to utilize convolutional neural networks CNN [43]. Many accessible platforms embraced AI techniques in their OCR architecture, such as ABBYFineReader [44] and Google Cloud Vision OCR [45] as they take raw images as an input, do the needed pre-processing and segmentation, and output text appeared in that image in an editable format. To evaluate the OCR performance, different measurements can be applied based on the supported languages (i.e Latin text versus multilingual text). Some of the common methods used to assess OCR performance are: 1) **Levenshtine distance** [46], which compares the truth ground text t with the OCR's extracted text s considering the number of deletions, insertions, and substitution needed to transform s into t using Equation 6, 2) **word recognition accuracy WRA** which divides the number of correctly recognized words by the total number of words, and 3) **word error rate WER** which is computed by subtracting the WRA from 1 (i.e. zero WER represents the best case scenario).

$$lev_{t,s}(i,j) = \begin{cases} max(i,j) & if min(i,j) = 0, \\ min \begin{cases} lev_{t,s}(i-1,j)+1 \\ lev_{t,s}(i,j-1)+1 \\ lev_{t,s}(i-1,j-1)+1_{(t_i \neq s_j)} \end{cases} & otherwise \end{cases} \tag{6}$$

Levenshtein distance, word recognition accuracy (WRA), and word error rate (WER) are some of the common methods used to asses OCR performance and these methods are detailed and used later on in this research in Section 6.

2.2.2 Natural Language Processing

Once the text is extracted, it can be taken into Natural Language Processing NLP models, which manipulate natural language automatically, to classify it. The collection of words is called **document**, and the full dataset is called **corpus**. In NLP models, the full cycle consists of the following stages: feature extraction, classifier selection, and evaluation [47]. In the feature extraction stage, the purpose is to clean and convert the unstructured text sequences into a structured format. The cleaning steps usually include tokenization (break a stream of text into smaller units as individual words), removing stopwords (i.e. dispensable words like "is, and, after"), capitalization (turning every letter into lower case), removing special characters and punctuation, and spelling correction. In addition, Stemming and Lemmatization approaches are usually used with NLP problems in order to reduce morphological variations of words. The difference between them is that stemming removes the last few characters of a word and usually results in incorrect meaning and spelling, which is called Stem. On the other hand, Lemmatization considers context by converting the word to its meaningful base form, which is called Lemma. For instance, the words {changing, changed, and change} have the stem "chang" and the lemma "change".

Feature extraction can be done using several approaches, such as **Term Frequency TF** [48] and **Term-Inverse Document Frequency TF-IDF**. TF is the number of repetitions the term appears in a document. For example, if we have a document consisting of 100 words and the term "restaurant" appeared 33 times, the TF of the word "restaurant" is $TF(restaurant) = 33/100 = 0.33$. On the other hand, TF-IDF measures the significance of that term "restaurant" in the whole corpus by computing the IDF which is the total logarithm of the number of documents divided by the number of documents containing that term. For instance, if the corpus (i.e. the whole dataset) is 10 million documents, and 300k documents contain the term "restaurant", then the $IDF(restaurant) = log(10,000,000/300,000) = 1.52$ and hence the TF-IDF of the word "restaurant" is computed by multiplying TF by IDF which is in this example $TF - IDF(restaurant) = 0.33 \times 1.52 = 0.502$. This way, tokens (i.e. words) are summarized and their importance is recorded for each class to be used in the next phase.

The existing classifiers in the field of NLP vary starting from the traditional ones like Naive Bayes to the models that are based on deep neural networks like RNN [49]. In this research, Multinomial Naive Bayes MNB and Support Vector Machine SVM [50] are used. In general, the Naive Bayes classifier is computationally inexpensive and does not require a big amount of memory [47]. For k classes where $k \in \{c_1, c_2, ..., c_k\}$, the probability of a class c given a document d can be known using Equation 7 where n_{wd} is the number of repetitions the term w occurs in document d.

The probability of the observed word w given a class c can be computed as follows in Equation 8 where D_c is the entire training documents in class c, and k is the number of unique words in all training documents (AKA the vocabulary size) [51].

$$P(c|d) = \frac{P(c)\prod_{w\in d} P(w|c)^{n_{wd}}}{P(d)} \tag{7}$$

$$P(w|c) = \frac{1 + \sum_{d\in D_c} n_{wd}}{k + \sum_{w'}\sum_{d\in D_c} n_{w'd}} \tag{8}$$

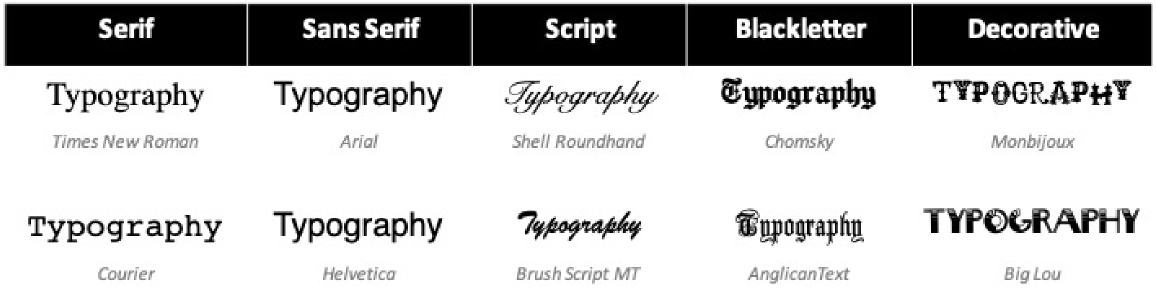
SVM is another common technique and one of the most efficient for document classification that uses a discriminative classifier by utilizing support vectors (i.e. data points with a minimum distance to the line that separates classes (hyperplane)). It works by mapping data to a high-dimensional feature space even when data are not linearly separable by using kernels. Although it is basically designed for binary classification problems [50], some techniques can be used to allow multi-classification like the One-vs-One technique, which builds one binary SVM for each class, and the One-vs-All technique [52], which builds one SVM classifier that attempts to distinguish a class from all other classes (i.e. either class X or all other classes). In this research, the second technique (One-vs-All) was used because the feature extraction method depends on TF-IDF.

For evaluation, the same performance measures used for object detection can be used for text classification. In particular, recall and precision, which are detailed in Section 2.1.2 and Equations 5 and 4, are used to evaluate text classifiers. In addition, **F-measure** (AKA f1-score) is calculated from recall and precision as elaborated in Equation 9, where 1.0 is the highest possible value that indicates both perfect precision and recall. F1 is the most valuable metric when it comes to imbalanced data as it is not affected with that imbalance [53].

$$F1 - score = \frac{TruePositive\ TP}{TruePositive\ TP + \frac{1}{2}(FalsePositive\ FP + FalseNegative\ FN)} \tag{9}$$

2.3 Typeface Design

Typeface is defined as the overall design of lettering which has a specific style. In particular, a set of fonts with common design features are called typeface. Type categories can be simplified to five main categories: Serif, Sans Serif, Script, Blackletter, and Decorative. Examples of these categories are shown in Figure 2.9. **Serif** characterized by small projections vertical to/angled at letters' terminals [27] while **Sans Serif** typefaces lack serifs. **Script** typefaces are imitating handwriting with connected script usually. **Blackletter** typefaces are featured by a dense, blackish texture, and decorated caps [54]. Finally, **Decorative** typefaces are meant to be distinctive and eye-catching, and their design is very broad with less rules to follow. They are also known as Display typefaces as they are usually used in big titles and headlines. In our research, we have considered these main categories in the annotation process for shop signboards.



Serif	Sans Serif	Script	Blackletter	Decorative
Typography Times New Roman	Typography Arial	Typography Shell Roundhand	Typography Chomsky	TYPOGRAPHY Monbijoux
Typography Courier	Typography Helvetica	Typography Brush Script MT	Typography AnglicanText	TYPOGRAPHY Big Lou

Figure 2.9: Main typeface categories with examples (font's name is written below)

Different typefaces have different design features including width, boldness (weight), contrast, and spacing. Some of these features make it hard for human to read. Likewise, some are challenging for machines to recognize. Therefore, choosing the right typeface for a shop signage is a vital step in order to achieve considerable degree of readability.

Chapter 3

Literature Review

In this chapter, all works related to this research will be reviewed including the existing storefront datasets in Section 3.1, the studies on storefront detection and classification in Section 3.2, license plates detection and recognition in Section 3.3, text extraction and recognition in scene images in Section 3.4, text classification using NLP in Section 3.5, and the influence of typeface design in Section 3.6.

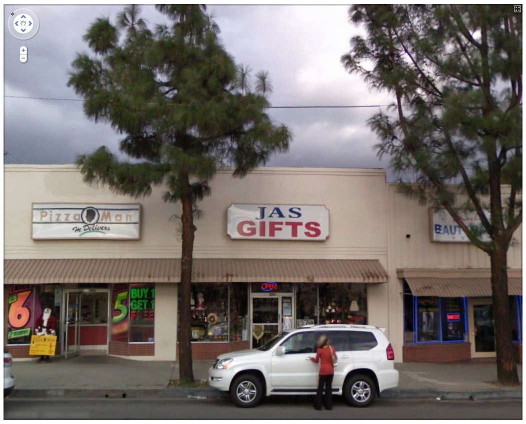
3.1 Storefront Datasets

Street View Text SVT dataset [10] is one of the earliest public datasets that focused on the signage of retail business. It consists of street view images that include storefronts from Google Map[1]. These images had been harvested using Amazon's Mechanical Turk service[2]. The dataset has 350 images in which each image has a single store signboard at least that includes some text. the total number of words in these images is 725. A set of one hundred images with a total of 211 words goes for the training set, and the rest for the test set. The Annotation for each image has been done by Alex Sorokin's Annotation Toolkit[3] which generates an XML format. For each image, business name and address are recorded in addition to image resolution, lexicon, and bounding boxes of words in the signage. The lexicon words have been collected from the top 20 business results that

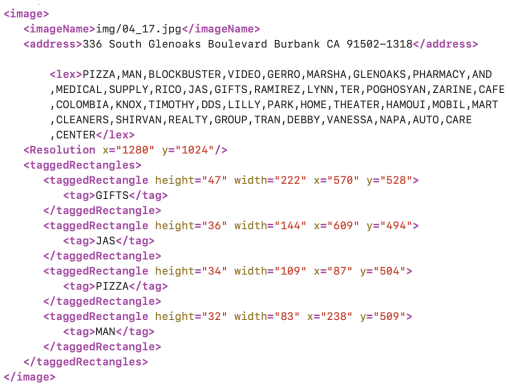
[1] http://maps.google.com

[2] http://mturk.com

[3] http://vision.cs.uiuc.edu/annotation/



(a)



```
<image>
   <imageName>img/04_17.jpg</imageName>
   <address>336 South Glenoaks Boulevard Burbank CA 91502-1318</address>

       <lex>PIZZA,MAN,BLOCKBUSTER,VIDEO,GERRO,MARSHA,GLENOAKS,PHARMACY,AND
       ,MEDICAL,SUPPLY,RICO,JAS,GIFTS,RAMIREZ,LYNN,TER,POGHOSYAN,ZARINE,CAFE
       ,COLOMBIA,KNOX,TIMOTHY,DDS,LILLY,PARK,HOME,THEATER,HAMOUI,MOBIL,MART
       ,CLEANERS,SHIRVAN,REALTY,GROUP,TRAN,DEBBY,VANESSA,NAPA,AUTO,CARE
       ,CENTER</lex>
   <Resolution x="1280" y="1024"/>
   <taggedRectangles>
      <taggedRectangle height="47" width="222" x="570" y="528">
         <tag>GIFTS</tag>
      </taggedRectangle>
      <taggedRectangle height="36" width="144" x="609" y="494">
         <tag>JAS</tag>
      </taggedRectangle>
      <taggedRectangle height="34" width="109" x="87" y="504">
         <tag>PIZZA</tag>
      </taggedRectangle>
      <taggedRectangle height="32" width="83" x="238" y="509">
         <tag>MAN</tag>
      </taggedRectangle>
   </taggedRectangles>
</image>
```

(b)

Figure 3.1: A sample from the SVT dataset [10] (a) image sample; (b) image annotation

came out from searching all nearby business stores which ended up in 50 unique words approximately. Figure 3.1 shows a sample of SVT images and their annotations in XML format.

Another Dataset [13] was collected also using Google Street View with Google collaboration. Unfortunately, the dataset is not provided to the public even after contacting the authors. The dataset contains 1.3 million street images with geo-information. They were collected from different urban areas in many cities across Europe, Australia, and the Americas. The annotation was done through some operators who were asked to label these images by generating bounding boxes around business-related information including signboards. In this study, the authors tend to multi-label street view storefronts, so they have 208 unique labels. They divided the data into the following: 1.2 million for the training set and 100,000 for the testing set. The splitting is location-aware to ensure that the business stores in the test set have never been seen in the training set; and to have samples in the training and the test sets from the same regions.

In addition to the previous datasets, a Chinese shop signs dataset (ShopSign) [55] has been published for the public. It contains images of real street views for shops signboards. The images were collected by 40 students for more than two years using 50 different types of smart phones and cameras. The ShopSign dataset has 25,770 Chinese shop sign images with geolocations which include a total of 4,072 unique Chinese characters (i.e. classes) with 626,280 occurrences. The annotation was done manually in a text-line-based manner with help of 12 faculty and graduate students using quadrilaterals. The dataset is large in scale, and it is considered sparse and imbalanced which increase its challenging level as the authors claimed. Moreover, the ShopSign dataset has big environmental and material diversity. To explain, some images were taken at night and under different lighting conditions. Also, five categories of shop signs were defined as “hard”: mirror, wooden, deformed, exposed, and obscure. Finally, the data has been split into the training and testing sets with 20,738 and 5,032 samples respectively. Unfortunately, this dataset is based on Chinese characters with limited samples of Latin characters.

Similarly, the Bangladesh Street View Signboard Objects BSVSO dataset [56] has been collected using Google Street View from 9 cities in Bangladesh where text is written in both Bengali and English. The dataset provides about 5k signboards but only 2600 are labeled. The annotations include bounding boxes information for detection purposes and they are available in XML and CSV files based on Pascal VOC format.

Unlike the above datasets which were collected mostly for shop storefronts, there are other available datasets for text in the real world. For example ICDAR2013 [57], ICDAR2015 [58], COCO-Text [59], and CTW [60] are some of the well-known datasets for text localization and recognition purposes. Also, Uber-text [61] is one of the large-scale datasets that has about 110k real-street images. Text within those images were hunted using polygons, and categorized into several categories including store name, street name, and licence number. The focus in these datasets is on detecting or recognizing the text itself. Hence, the images were not annotated for store classification.

Many other well-known datasets are available for object detection in general, such as Pascal VOC [62, 63], ILSVRC [35], and MS-COCO [39]. They are usually used to evaluate object detection models. These datasets are much more general than the scope of this research. This prevents the usage of these datasets in this research as they require manual re-annotation and filtration to extract the required information. However, it is possible to use the models that were pre-trained on one of these datasets, which is also known as transfer learning.

	SVT DS	Google DS	ShopSign DS	ShoS DS (ours)
Type	Public	Private	Public	Public
Source	GSV	GSV	Real SV	GSV
How collected	Amazon Mechanical Turk	Locally from Google data	Manually	Upwork freelance platform
Where	unknown	Europe, Australia, Americas	China	United States, Canada
Images	350	1.3m	25,770	10k signs within 7500 image
Annotation	Alex Sorokin's Toolkit	Operators	Manually	VGA Image Annotator+ operators
What	Words in nearby businesses	Whole storefront + labels	Characters in store signboards	store signboards + store classes + words + typeface class + difficulty level
Classes	∼50 lexicon words per image	208 store labels	4,072 unique Chinese characters	7 store classes 6 typeface classes
Language	Latin	Latin	Chinese mainly	English

Table 3.1: Summary of the storefront datasets compared to the ShoS dataset

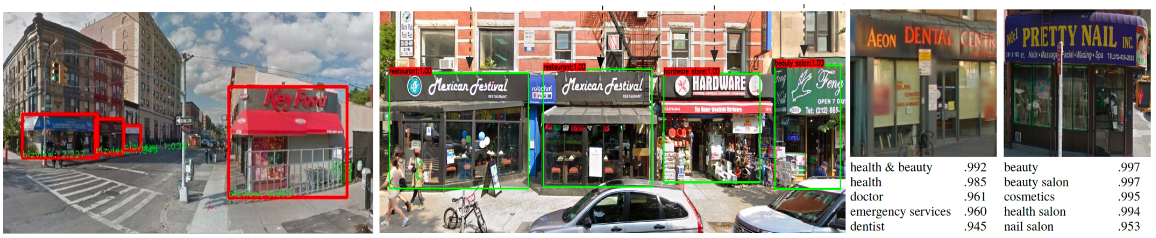
Table 3.1 summarizes the most comparable datasets to ours (the ShoS - detailed in Chapter 4). They are either private or limited in their features which motivate us to provide the ShoS dataset.

3.2 Storefronts Detection and Classification

In [11] study, the proposed model detected the whole storefronts given street panoramic views using MultiBox model [41]. MultiBox uses a single CNN based on GoogLeNet [64] with a 7×7 grid. To make less number of predictions of bounding boxes, the authors applied a coarse sliding window fashion with a small overlap threshold (minimum 0.2). Then, a post-classifier that is based on GoogLeNet and implemented in the RCNN manner, was used followed by non-maximum suppression to get the final score of detection. During training, the original input image size was reduced by a factor of eight for both models. In addition, some negative samples, proposals with low confidence from the MultiBox model, were used in training the post-classifier model. This work used almost the same large-scale dataset used in [13], however it found difficulty annotating the huge number of the dataset, so less amount of data was used in testing. For evaluation, the authors compared their work with selective search and multi-context heat MCH map. Results revealed that MultiBox surpasses the other methods with a recall of 91% compared to 62% for selective search. Unfortunately, MCH could not detect the boundary of storefronts precisely. This was because the fact that storefronts are more exposed to noise and they can abut each other.

Similarly, another recent study [12] proposed a system composed of several models for detecting the whole storefront from street views and classify it in further models. The detector was based on YOLOv3 [6], and its output was fed into a classifier. The classifier extracts morphological and textual information and uses them as cues. The work was evaluated using a very limited dataset, and their methodology got mAP@0.50 = 79.37% for detection and 80.44% for classification involving textual information. Moreover, in [56], the target was to tackle the overlapping objects within store signboards in street view images. Variant image color schemes had been tested using Faster RCNN to detect signboards. The findings showed that P_RGB color scheme suppressed the others with a mAp@07=80%.

In [13], the abundance of Google street views was leveraged to train a CNN model based on GoogLeNet backbone [64] for storefront multi-label classification. After annotating (as described in section 3.1) by some operators, the authors noticed some inconsistency in the labelling (i.e. McDonalds can be labelled as fast-food, hamburger, and take-away restaurant) and that confuses the learner. So, they built their model using ontological classifications in which a group of labels belongs to a high-level class (restaurant in the previous McDonalds example). So, during training, textual information extracted by an OCR was used beside the geo-information to match the extracted batches with the ground truth. In testing, each image was associated with one or more labels sorted by classifier's confidence. So, the top k predictions were chosen and compared with the ground truth. The result was considered correct if the intersection between the ground truth set and the top k prediction set is not empty. By doing so, 83% accuracy was achieved. Moreover, the authors compared their model results with human performance after conducting two large scale surveys for model evaluation. They found that their model was close enough to human-level accuracy. Finally, the text appeared on store signboards were investigated to see its influence on the model. That was done by feeding two groups of the same sample images: one with blurred text, and the other with clear text. The outcomes showed that the model learns from text as it gave better results when text was not blurred. That emphasizes the importance of having readable text by machine.



(a) (b) (c)

Figure 3.2: Samples from the literature work for (a) storefront detection [11], (b) storefront detection and classification[12], and (c) store multi-label classification[13]

Other studies [65, 53] based their work on extracting features from the open source maps and the available information on social media platforms. Features were extracted from: map tags [65], operation-based, review-based, neighborhood-based, topic-based, and visual attributes [53] to classify or tag points of interest using machine learning techniques. The problem is, some of the targeted information for the feature extraction might be missing or incorrect. For example, it is mentioned in both studies that the operating hours for a point of interest were the most powerful feature to differentiate between a restaurant and a bar for example. Though, that would not be the rule for other types of business stores that were not included in the scope of the mentioned studies.

Most of the previous mentioned works used the whole storefront in their systems (see Figure 3.2), and faced some crucial issues in detection and classification because of 1) the limitation of existing datasets as detailed before in Section 3.1, 2) the boundaries of storefronts are not clear enough to be learned, and 3) some irrelevant information that can be found in the storefront may mislead the classification process. That motivated us to tackle such issues by focusing on store signboards as they present much more consistent appearance and include semantic textual information useful for store classification.

3.3 License Plates Detection and Recognition

License Plate Detection LPD is the area most similar to our work, and it has gotten a lot of attention with the enhancement of real-time object detectors. In [66], the first two models of YOLO [31, 33] were used to detect the vehicles first and then the license plate. The box with a higher confidence score is considered. They were able to get high precision and recall near 99%. In addition, in [67], a YOLO-inspired solution for LPD system were introduced with different hyper parameters. A recall ratio of 98.38% was achieved outperforming two commercial products: OpenALPR and Plate Recognizer significantly.

Another work [68] focused on providing a small and fast model for LPD to employ it on embedded system. The proposed system was based on Mobilenet-SSD MSSD to detect license plates. The authors further optimized the system by introducing feature fusion methods on the MSSD model in order to extract context information and hence better detection. The results reveal that their proposed system is 2.11% higher than the MSSD in terms of precision, and it is also faster than MSSD by 70ms.

Similarly, the authors of [69] leverage SSD [9] detector to build a model for LPD. First, SSD detects vehicle regions. Next, these regions were cropped, and the character candidates were generated using maximally stable extremal regions MSER algorithm with some thresholds to eliminate false character candidates. Then, nearby characters were grouped by a bounding box based on specified spatial distance and other factors. Finally, a filtration process is done for the extracted license plates by comparing the dimension of the detected vehicle and the plate(s) candidates. Finally, a CNN net with MobileNets architecture provided by [70] was used to classify word/no-word classes. The proposed system has been evaluated in terms of precision and recall, and it reaches significant results even with high IoU (=0.7) as it gets 96.88% and 98.41% respectively.

Despite the significant enhancement of LPD systems, things could be different with shop signboards as they take various shapes and forms compared to license plates. This visual variability may make it challenging for machines to detect and classify them.

3.4 Text Extraction and Recognition

End-to-end systems were targeted in this research to detect the text and recognize it (i.e. transform it into machine readable and editable text). Such systems are known as Optical Character Recognition OCR. Extracting text from scene images differs from document images as the first one would have many challenging factors, such as noise, occlusion, and variation. Applying traditional methods on scene images may result in a lot of rubbish and false alarms. On the other hand, deep learning based systems would outperform traditional approaches by a significant margin [71, 72]. Many studies [73, 74] built end-to-end OCRs employing deep learning algorithms, such as CNN, RNN, and DNN. The focus of this research will be on AI-based OCRs that are available either as on premise solutions or as client-server interfaces. This is because these kinds of tools are already trained on a tremendous amount of data and ready for end-to-end use. Some examples of these existing tools are ABBY FineReader [44], Tesseract[4], Google Docs OCR[5], Google Cloud Vision xxs[45], Microsoft oneNote, Readiris[6], and Transym[7].

In [75] four OCR tools, Google Docs OCR, Tesseract, ABBY FineReader, and Transym, were tested and comparatively evaluated using a dataset that contains 15 different categories including noisy images, multi-oriented text, and machine/handwriting images from natural scenes. Results revealed that Google Docs and ABBY FineReader perform the best among the others even with hard conditions like noisy, blurred, and skewed text. The accuracy reached 74% and 71% for Google Docs and ABBY FineReader respectively. When brightness and contrast enhancements were applied to images the results did not improve much. Similarly, in [76] work, Tesseract, ABBY

[4] https://github.com/tesseract-ocr/tesseract
[5] https://docs.google.com/
[6] https://www.irislink.com/
[7] https://transym.com/

Finereader, and GOCR were evaluated using NEOCR dataset [77, 78], which contains text from the natural environment, considering different domains including fonts, orientation, and blurriness. The total average Levenshtine distance scores for all of the studied domains showed similarity between Tesseract and ABBY Finereader as they scored 1.8 and 1.7 respectively (the lower the better). However, only ABBY Finereader performed better with "special font" under the font domain. Furthermore, the proposed work in [79] compared ABBY Finereader and Tesseract on MID-500 dataset [80], which contains ID images taken with mobile devices under several conditions. The findings showed that the average per-character recognition rate PCR for ABBYY Finereader v15 is better (60.91%) than Tesseract v4 (55.75%) for all the studied text line types.

Google Cloud Vision OCR is an open source API that allows the integration of android mobile devices. A study [81] built an android app to extract text from business cards, posters, flyers, and magazines using Google Cloud Vision OCR. Then, the useful information like names, dates, and locations were extracted utilizing NLP techniques. They tested the app on their dataset, so the OCR outcome was considered correct when all the characters in an image were correctly identified. Without any image pre-processing Google Cloud Vision OCR reached an average accuracy of 75.25% for all image types. Another study [82] intended to investigate the robustness of the Google Cloud Vision project including OCR. The observations revealed that it is highly robust especially with the solid background and text. Last but not least, another study [83] evaluated two of the off-the-shelf OCRs, Google Cloud Vision OCR and Microsoft Cognitive Services[8], on elements that combined textual and graphical components. The results showed that Google Cloud vision OCR outperformed the Microsoft Cognitive Services. Also, most of the errors occurred by Google Cloud vision OCR were because of lacking of some special characters support like the ones used in mathematical formulas.

From previous studies, it had been chosen for this research two of the cloud services OCRs ABBY FineReader and Google Cloud Vision OCR as they stand out among the other reviewed OCRs.

[8] https://azure.microsoft.com/

3.5 Text Classification using NLP

Categorizing a group of documents into pre-defined classes is called supervised classification [84]. The field of text classification in natural language processing NLP is one of the well-studied areas for decades using machine learning. The era of deep learning also added more advancement for the field, however in this research we applied some of the traditional machine learning techniques and avoid deep nets for the following reasons [47]: 1) text classification based on deep learning algorithms requires a tremendous amount of data for training compared with traditional methods, 2) the computational cost for deep learning approaches might get complicated during training and, 3) the limitation of a comprehensive theoretical understanding of the learning process increases because of the concept of “blackbox” nature in deep learning methods.

A comparative study [85] applied some of the text classification traditional methods to classify news into six different categories. Naive Bayes NB, K-nearest neighbor KNN, and SVM were tested on data collected from public news like CNN, and Fox News where each class has 30 documents. Several hyperparameters were implemented for the stemming techniques. Results showed that KNN and SVM performed better while NB was in the average range. Similarly, another work [86] compared Multinomial NB, KNN, and Decision Tree DT for topic classification which considers six classes. Data were collected from Amazon’s product reviews with 6k documents in total. The outcomes disclosed the superiority of MNB among the others with an F1-score reached 91.8%. Moreover, Recurrent Neural Network RNN, SVM, and Multinomial NB were comparatively studied to classify spam in emails [87]. The data were acquired from Kaggle with a total of 5k emails approximately. In terms of F1-score, SVM was the best followed by MNB with 94% and 85% respectively.

Considering the previous works, it is clear that traditional methods still perform better in some of the text classification problems. Therefore, it had been decided for this research to utilize two classical classifiers and the are two of the best working methods: MNB and SVM.

3.6 Influence of Typeface Design

Although plenty of studies have been conducted in order to test legibility and readability of some fonts, they were tested from the human side only as a reader, such as in [88, 89, 90]. Typefaces used in street signage fall in a totally different environment. In addition to the materials they are composed of, their appearance is dictated by the distance between them, which can range from a few feet to hundreds of yards. Their readability by human is also different than machine recognition. Few studies have focused on the influence of type design on recognition systems. In [23], the authors studied the impact of font design on vehicle license plate detection system. Two fonts were tested: Mandatory, and Driver Gothic. Hence, some confusion cases were found. For example, the I-1, Q-O, and 0-O cases reported high confusion due to the similarities in geometric features. Also, their experimental analysis discovered some severe cases in Mandatory font like when the system is confused between letter I and digit 1 because of their identical glyph. Another recent study [91] investigated the effect of high-stroke-contrast fonts (i.e. contrast between thicker and thinner parts of a letter) on reading as they are used usually by designers to look more fashionable. three types of contrasts were tested: high contrast, no contrast, and in between. Results assured that it is better to avoid high stroke contrast when letter recognition is a priority.

Multiple factors may affect the choice of fonts used on street signs in general. Some of them follow the surrounding environment especially for urban cities [26], and some stick to the traditional sign-writing which is close to inscriptions [15]. Many other decisions regarding the type of font used on signboards might be based on psychological studies like [92, 93]. They suggested using human-like typefaces in products, menus, and even advertisement signage as they would bond them together and increase customers' attachment to the brand. All these factors may create a challenge for machine recognition. Thus, a further investigation was done in this research to see the influence of different typeface styles on the recognition process.

Chapter 4

Dataset: The ShoS

Literature in Chapter 3 highlights the lack of public datasets for business storefronts. In the following sections, detailed information about the stages of collecting and annotating the new data is provided in Sections 4.1 and 4.2. Also, revising and cleaning the collected data were detailed in Section 4.3. In addition, the way of building a store class dictionary is described in Section 4.4. Furthermore, challenges and limitations related to the ShoS dataset are discussed in Section 4.5.

4.1 Data Collection

The **Shop Signboard Dataset ShoS** has been collected using Google Street View GSV. This version of the dataset was collected from 51 cities in Canada and the USA including Toronto, Vancouver, Ottawa, Calgary, Edmonton, Chicago, Los Angeles, San Francisco, New York City, Seattle, Miami, and Boston. Using GSV, screenshot images of storefronts were taken after some adjustments to ensure the clarity of the image. The recorded scene includes one or more shop signboards with a minimum of one store per image and a maximum of 7 stores per image (Figure 4.1). The average number of stores per image is 1.6 and more statistical descriptives are provided in Table 4.1. View angles were selected in a way that guarantees signboard visibility. The samples were collected by the researchers at first and by hired freelancers later through Upwork[1] freelance platform. The completed process of collecting and annotating the dataset took around one year.

[1] https://www.upwork.com/

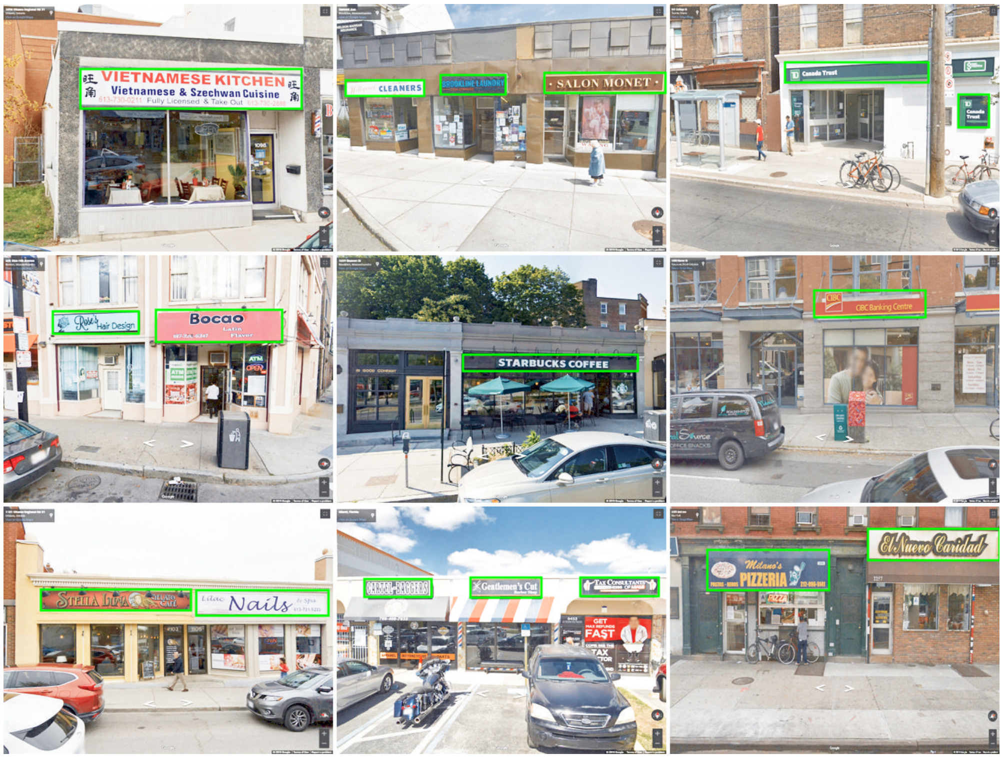


Figure 4.2: Sample images from the ShoS Dataset illustrated with the bounding box annotations

Descriptives	region_count
N	10000
Missing	0
Mean	1.63
Median	1.00
Standard deviation	0.844
Variance	0.712

Table 4.1: Statistical descriptives of the number of signboard (region_count) per image

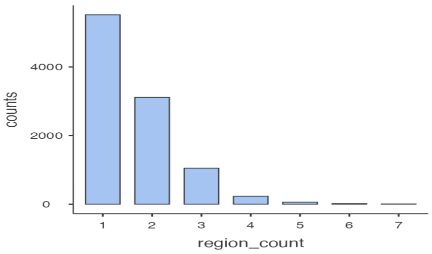


Figure 4.1: Bar chart of the number of signboard (region_count) per image

The ShoS dataset contains 10k signboards within 7500 images of multiple resolutions, mainly in 3360x2100 and 1280x1024. The shop signboards were cropped out of the full street view generating another dataset named **the ShoS-cropped** to enable usage of both datasets for multiple research purposes. Figures 4.2 and 4.3 show some sample images from the datasets.

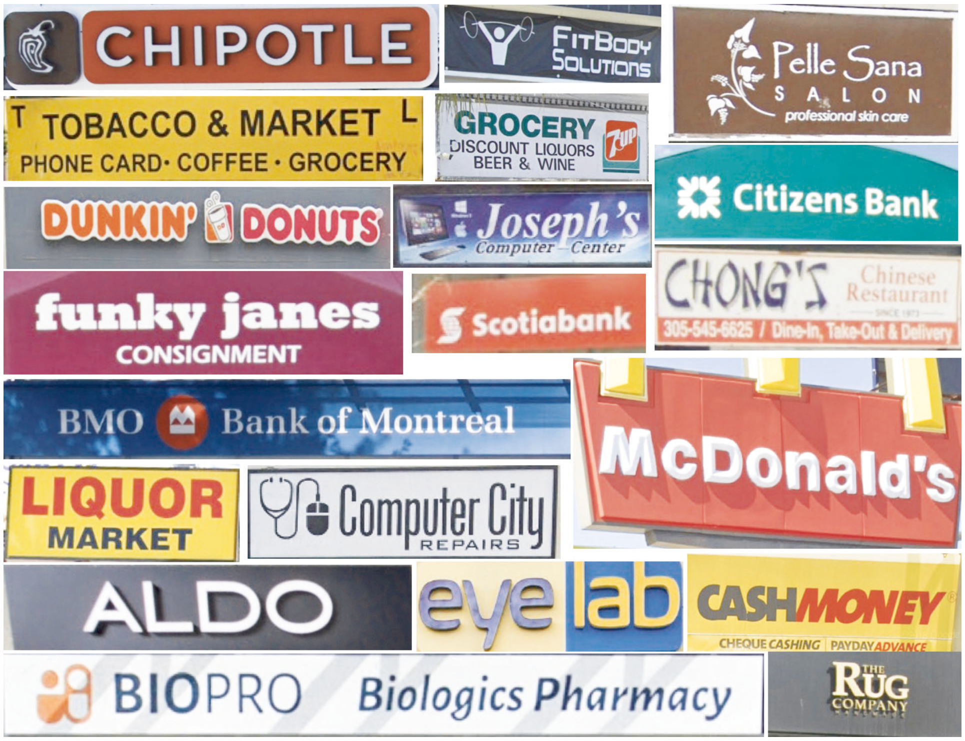
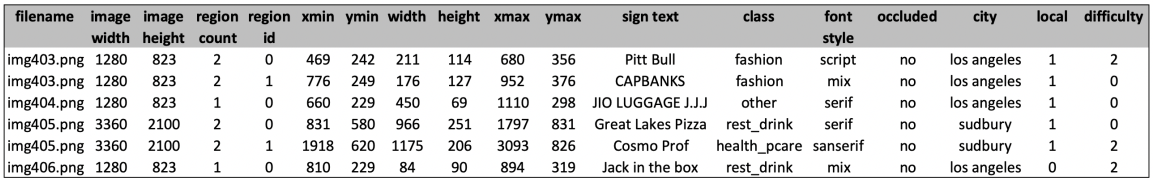


Figure 4.3: Sample images of the ShoS-cropped Dataset

4.2 Data Annotation

There are several open source image annotation tools. This research used VGG Image Annotator (VIA) [94, 95] which was developed by Visual Geometry Group (VGG) as a project under the Department of Engineering Science at the University of Oxford. VIA is a manual annotation tool that is characterized by its simplicity and lightweight. It does not need any installation, and it works online with an interface or offline as an HTML file. VIA satisfied the requirements of this research as it enables adding more attributes to image files and bounding boxes where many tools do not. Also, it can generate annotation files as JSON and CSV files. However, additional modifications to the CSV file were needed to meet the research requirements. The final CSV file was then used to generate Pascal VOC XML and text files for each image through Python scripts.

For each image file, the following attributes are recorded: image name, image width, image height, and the number of bounding boxes in the image. Similarly, for each bounding box in each image, these attributes were annotated: top left coordinates (xmin, ymin), bottom right coordinates (xmax, ymax), bounding box width, bounding box height, text inside the signboard excluding numbers and addresses, store class, font style, local or chain, occluded or not, difficulty, and the city that the image was collected from. Figure 4.4 shows few samples from the CSV annotation file for the ShoS dataset.



filename	image width	image height	region count	region id	xmin	ymin	width	height	xmax	ymax	sign text	class	font style	occluded	city	local	difficulty
img403.png	1280	823	2	0	469	242	211	114	680	356	Pitt Bull	fashion	script	no	los angeles	1	2
img403.png	1280	823	2	1	776	249	176	127	952	376	CAPBANKS	fashion	mix	no	los angeles	1	0
img404.png	1280	823	1	0	660	229	450	69	1110	298	JIO LUGGAGE J.J.J	other	serif	no	los angeles	1	0
img405.png	3360	2100	2	0	831	580	966	251	1797	831	Great Lakes Pizza	rest_drink	serif	no	sudbury	1	0
img405.png	3360	2100	2	1	1918	620	1175	206	3093	826	Cosmo Prof	health_pcare	sanserif	no	sudbury	1	2
img406.png	1280	823	1	0	810	229	84	90	894	319	Jack in the box	rest_drink	mix	no	los angeles	0	2

Figure 4.4: Some samples from the CSV annotation file for the ShoS dataset

Store Classes were chosen based on the North American Industry Classification System NAICS. However, the research scope was limited to six super-classes defined as the following:

(1) **Restaurants and Drinking** which includes full/limited services restaurants, fast food restaurants, coffee houses, and bars.

(2) **Food and Beverages** which includes grocery stores (supermarkets and convenience stores), and specialty food stores (meat markets, fish markets, and fruit/vegetables markets).

(3) **Health and Personal Care** which includes clinics, pharmacies and drug stores, optical good stores, cosmetic/beauty supplies and perfume stores, and GYM.

(4) **Finance and Investing** which includes banks, insurance companies, money marts, accounting and tax services, and real estate.

(5) **Technology** which includes computer/electronic stores, telecommunication stores, and digital printing and copying services.

(6) **Fashion** which includes clothing stores, jewelry/accessories stores, and shoe stores.

Any remaining category goes under the “Other” class. The research defined sub-classes under each class to precisely identify which signboard belongs to which class. Figure 4.5 lists the hierarchy of super and sub-classes considered in the ShoS dataset, and Figure 4.6 illustrates their

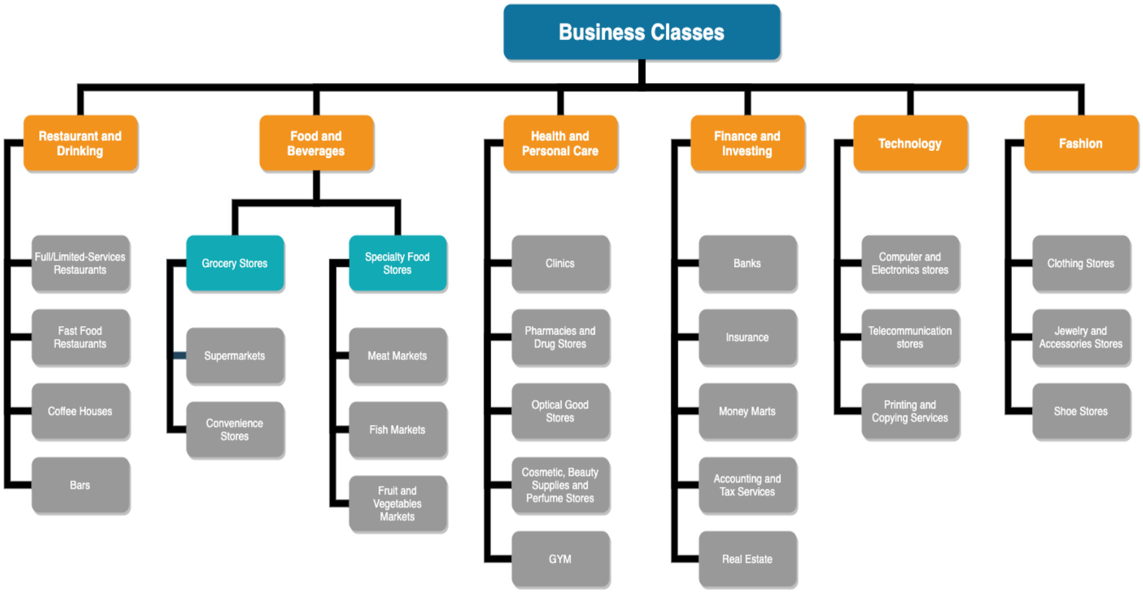


Figure 4.5: The research scope of store classes considered in the ShoS dataset

distribution in the ShoS dataset.

The **font style** attribute identifies the main typeface styles (Serif, San Serif, Script, Decorative, and Blackletter) in addition to Mix for text with mixed styles. Examples of these typeface categories are shown in Figure 2.9 and their distributions in the ShoS dataset are illustrated in Figure 4.7.

The attribute **local** checks whether the store is a chain store (value=0) or a local store (value=1). The chain stores are identified as one of several stores in multiple locations sharing the same brand and central management. This is to ensure that the ShoS dataset is not biased towards chain stores. The ShoS dataset has 84% local stores. Figure 4.8 illustrates the ratio of local versus chain stores in general and for each class in the ShoS dataset. It is clear that class Finance and Investment has a higher ratio of chain stores compared to the remaining classes and that is due to the dominance of large banks in the collected data.

A difficulty level for each signboard in the ShoS dataset had been determined based on multiple factors to reflect real world situations. First, occlusion, which is represented in the attribute

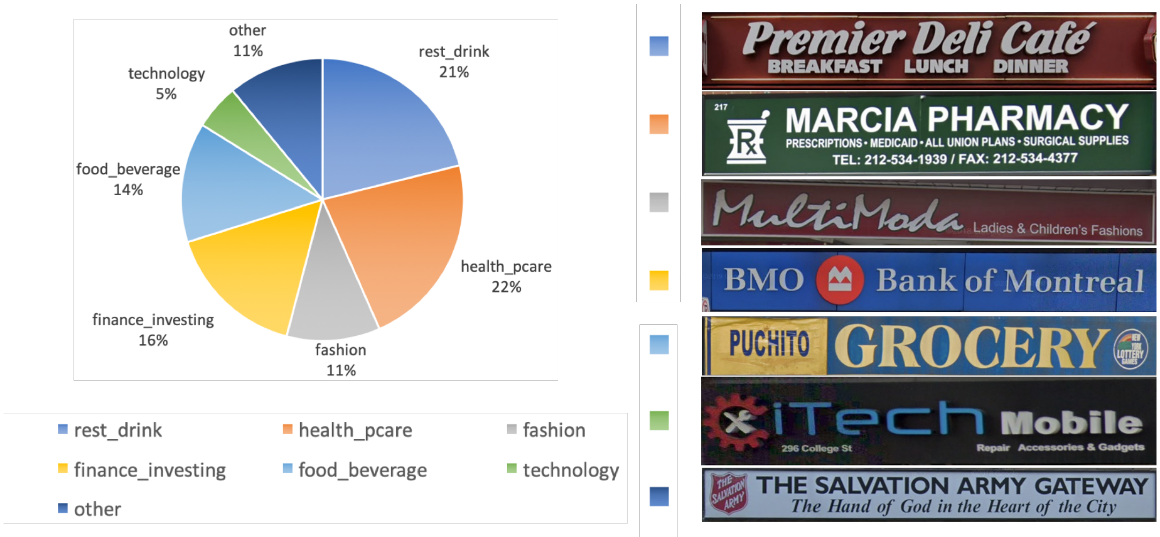


Figure 4.6: Distribution of store classes in the ShoS dataset along with a sample image of each class

occluded, where shop signboards can be occluded by trees, traffic signs, big vehicles, spotlights, shadows, and some other shop signs that are positioned for pedestrians (see Figure 4.9 for examples of occluded signboards). The signboard is considered occluded if at least 20% of the text on the sign is covered. Around 9% of the ShoS signboards are occluded. Second, a rating system had been utilized to evaluate the difficulty level based on environmental and confounding factors in attribute **difficulty**. To explain, image-related issues, such as bad angle, unclear or blurred view, and bad resolution were assigned the value=1. The second difficulty scale, with the value=2, was assigned to store names that do not indicate any semantic clue regarding its type. Another difficulty scale defined with the value=3 was assigned to stores with misleading names. This difficulty level was assigned when terms that are not related to the store class were used intentionally in a creative manner or because the store is selling products under a different category than its main class. For example, a grocery store, that falls under "Food and Beverage" class, added the terms "jewelry, and hair products" to its signboard which might be confusing for shop classification. Finally, storefronts that are crowded with advertisements and non-signage elements were assigned the difficulty value=4. About 16% of the ShoS signboard images are difficult which increases the challenge of the dataset. Table 4.2 provides more details and examples of the rating system in addition to the occurrence percentage for each difficulty level.

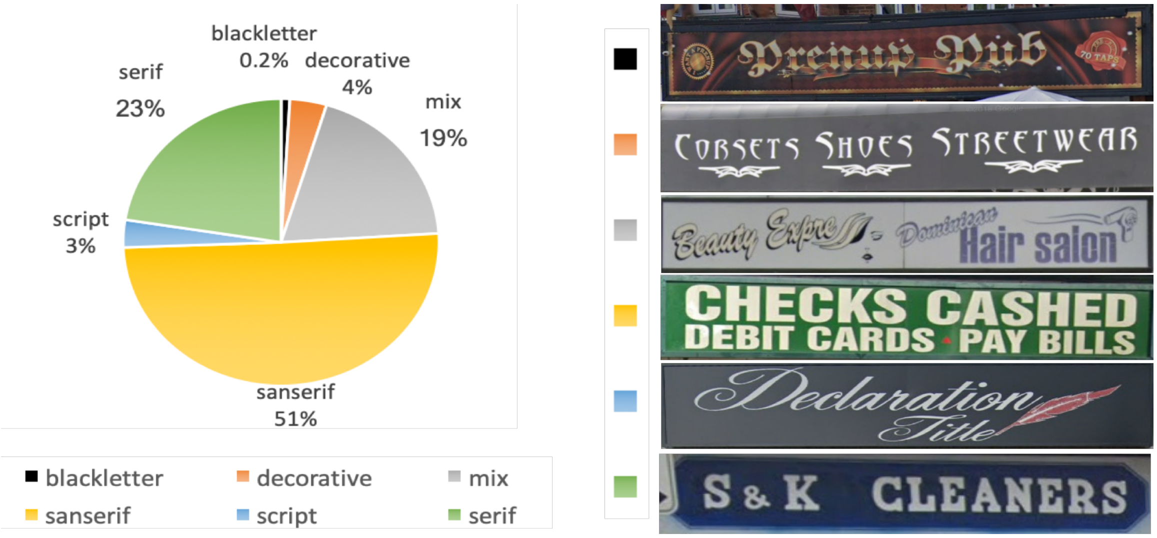


Figure 4.7: Distribution of typeface styles in the ShoS dataset along with a sample image of each style

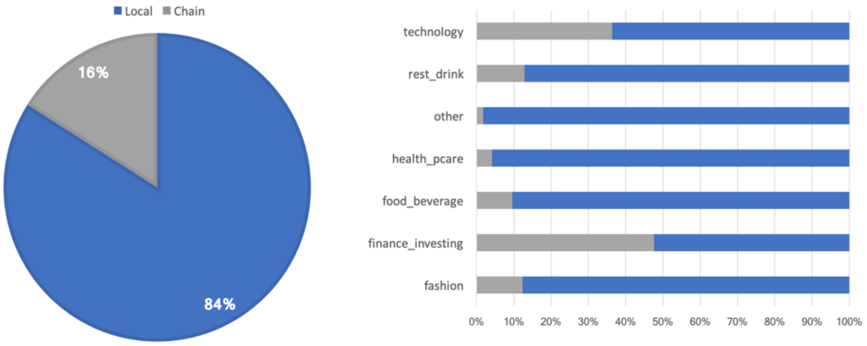


Figure 4.8: The ratio of local vs chain stores in general and for each class in the ShoS dataset

Finally, extra steps were applied to automatically generate bounding boxes of words within each signboard in the ShoS-Cropped dataset. The word-level annotation was produced in a separate CSV file. This had been done using Google Cloud Vision OCR [45] which automatically generates bounding boxes around each extracted word and around the whole text. Figure 4.10 shows some samples from the mentioned word-level annotation file and the corresponding illustration of the bounding boxes on the signboard image. About 53k words were annotated in a CSV format where each signboard image has multiple rows for each recognized word within that image associated with its top-left and bottom-right coordinates in addition to the entire extracted text. It is important to

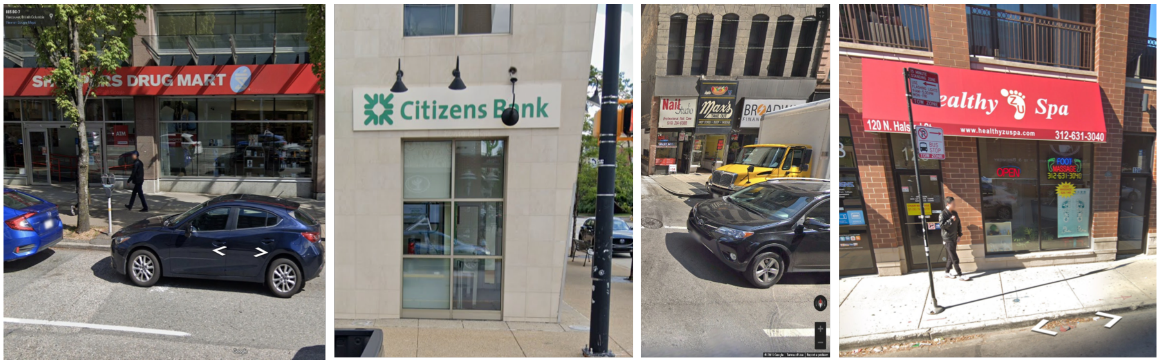
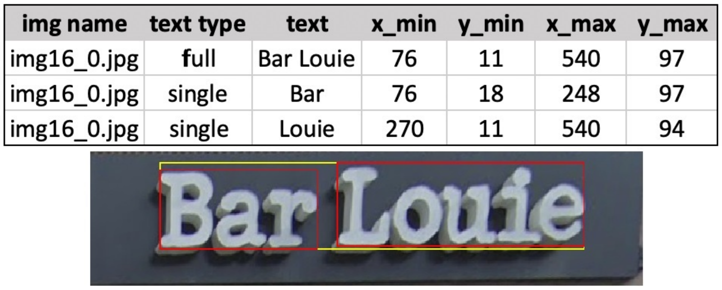


Figure 4.9: Examples of occluded signboards from the ShoS dataset

Difficulty	**Description**	**Occurrence**	**Example**
0	Not difficult	84.43%	-
1	Bad angle, unclear or blurred view, bad resolution	1.32%	
2	Text is not descriptive for the designated class	12.88%	
3	Misleading text, multi classes in one signboard	0.82%	
4	Storefront is crowded with ads and non-signage elements	0.55%	

Table 4.2: Elaboration of difficulty scales with example images from the ShoS dataset

mention that this word-level annotation is not extremely accurate as it depends on the OCR accuracy, so it might need to be reviewed by human operators to leverage it in recognition research.



img name	text type	text	x_min	y_min	x_max	y_max
img16_0.jpg	full	Bar Louie	76	11	540	97
img16_0.jpg	single	Bar	76	18	248	97
img16_0.jpg	single	Louie	270	11	540	94

Figure 4.10: A sample from the word-level annotation CSV file and its representation on the image where yellow bounding box surrounds the full text patch and the red ones show the annotation at word-level.

There was no necessity to utilize this word-level annotation in this research as the scope is different.

4.3 Data Revising and Cleaning

The ShoS dataset were revised and cleaned following specific protocols. First, the hired freelancers were provided with a guideline to follow during data collection, such as avoiding some board types like painted glass and illuminated signboards. An initial review was performed to accept or reject the collected images based on the given guidelines. Once accepted, the freelancers annotated the images based upon more detailed instructions. The annotated coordinates for each signboards were visualized to check their correctness and accept/reject them. Then, the "difficulty" and "local" attributes were annotated by the researchers at the end to ensure uniformity. After that, the annotation file was processed using a Python script to avoid null values and mismatched attributes. For example, a street view image could be recorded to have three storefronts but only two signboards were annotated. So, the image goes under individual investigation to fix the mismatching by either annotate the missing signboard or fix the number of region counts. Figure 4.11 shows the work flow of the revising and cleaning step of the ShoS dataset. Finally, the whole annotation file was revised manually over multiple stages to validate store class, font style, and sign text attributes.

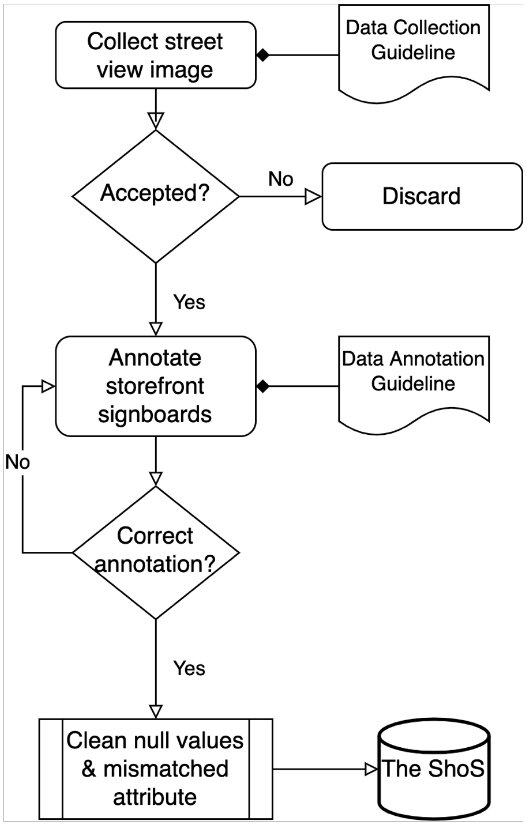
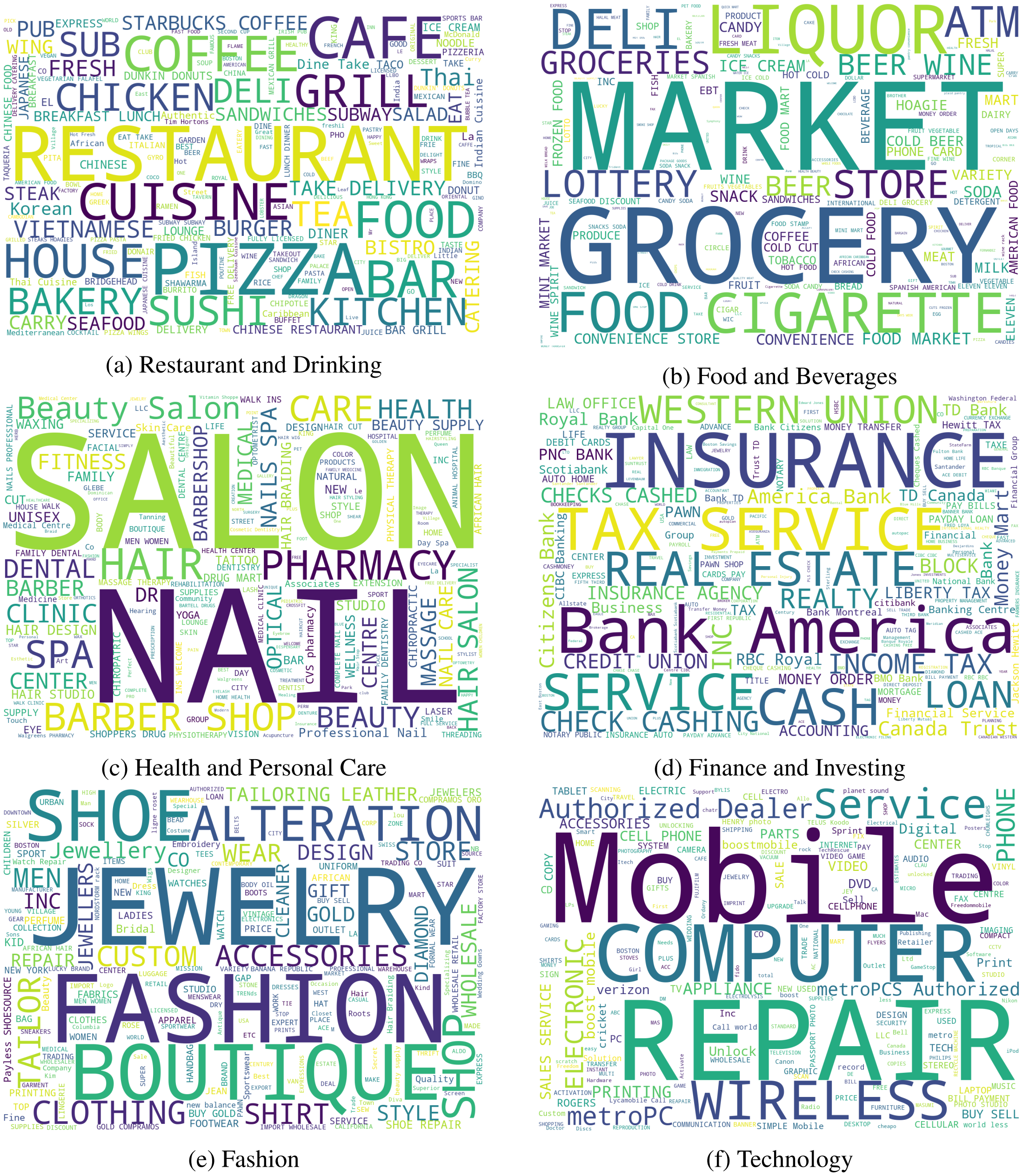

Collect street view image
Data Collection Guideline
Accepted?
No
Discard
Yes
Annotate storefront signboards
Data Annotation Guideline
No
Correct annotation?
Yes
Clean null values & mismatched attribute
The ShoS

Figure 4.11: A flow chart of the revising and cleaning step of the ShoS dataset

4.4 Shop Class Dictionary

A class dictionary was built based on the ShoS dataset annotations for each shop class to figure out the most frequent words which would be useful in the classification stage. Initially, the text data in all signboards was cleaned by converting it into lower case and removing all non-Latin characters, numbers, punctuations, brackets, stopwords, and all single-character words. Then, the frequency of each word for each class was counted, and the top 200 frequent words for each class were recorded. Next, the lists of frequent words were manually screened to eliminate words that are deemed non-predictive. Table 4.3 shows the top-15 frequent words in each class. Figure 4.12 visualizes the top 200 frequent words for each class in the ShoS dataset utilizing the *wordcloud* library in Python where bigger words represent most frequent ones.



(a) Restaurant and Drinking

(b) Food and Beverages

(c) Health and Personal Care

(d) Finance and Investing

(e) Fashion

(f) Technology

Figure 4.12: The word cloud from the top 200 frequent words of each class in the ShoS

rest_drink	food_beverage	health_pcare	finance_investing	fashion	technology
Restaurant	Market	Salon	Bank	Jewelry	Mobile
Pizza	Food	Hair	Insurance	Fashion	Repair
Cuisine	Grocery	Nails	Tax	Boutique	Wireless
Coffee	Beer	Beauty	Money	Gold	Computer
Bar	Deli	Spa	Financial	Shoes	Authorized
Grill	Wine	Barber	Loans	Clothing	Dealer
Food	Convenience	Pharmacy	Cash	Accessories	Electronics
Cafe	Cold	Care	Real	Wholesale	Phone
Takeout	Liquor	Dental	Estate	Alterations	Cell
Chinese	Mart	Health	Credit	Wear	Video
Chicken	Soda	Clinic	Check	Tailor	Printing
Sushi	Ice	Medical	Income	Custom	Digital
Thai	Candy	Family	Pawn	Watch	Appliance
Kitchen	Snacks	Massage	liberty	Beauty	Photo
Bakery	Lottery	Fitness	Agency	Bridal	Parts

Table 4.3: The top-15 most frequent words in each store class in the ShoS dataset

4.5 Challenges and Limitations

During data collection, multiple challenges were faced that limited the overall diversity of real-world store signboards, and they are detailed below:

A) Google Street View challenges. Some factors impact the clarity of shop signboards in Google Street Views GSV. That is, GSV collects street views by taking panoramic view images and stitching them together [96]. Stitching images can create irregular shapes with repetitive or incomplete parts. Moreover, environmental factors may affect the clarity of GSV images, such as rain which leads to foggy lens and thus foggy images. In addition, GSV images can be obstructed by traffic on the street, such as buses and large vehicles. All of these factors can obstruct shop signboards or impact the quality of their screenshots. Moreover, almost all of GSV images were originally taken during daylight. That limited the ShoS to daylight images only.

B) Signboard material challenges. Different materials can be used to build shop signboards including mirror, glass, wood, or even bare wall. The study in [55] identified some of these material types as ’hard’ for the detection and recognition process. Therefore, the number of signboards that are made of those material types was minimized from the ShoS dataset as much as possible.

C) Signboard position challenges. Some shops place their signboards in abnormal positions. For example, shop signage can be perpendicular to the storefront to attract pedestrians. Also, it is possible to find shop signboards on top of buildings without an actual storefront. Capturing these types of signboards could confuse the learning process as they do not reflect the normal view of a storefront with a signboard. Thus, such images were eliminated from the ShoS dataset.

D) Imbalanced classes. The ShoS and the ShoS-cropped datasets can be used for two different types of classification: shop classification and typeface style classification. Nevertheless, the classes are not balanced as they reflect the real world situation. For instance, class technology in store classification is a way less than other classes which reverberates the reality. Yet, this imbalance was handled utilizing some augmentation techniques as elaborated in Chapter 5. Also, the "Blackletter" typeface style is barely used in signboards as it is difficult to be read/recognized. Thus, fewer samples for such classes are available. Since there was insufficient amount of samples in the "Blackletter" typeface, it was eliminated in the statistical study done in Chapter 7.

4.6 Dataset Availability

The dataset will be made public for research purposes upon request from CENPARMI research manager Nicola Nobile at nicola@cenparmi.concordia.ca. In this research, the ShoS and the ShoS cropped datasets were utalized to train and test the models.

Chapter 5

Shop Signboard Detection and Classification Framework (SSDCF)

The main purpose of this research is to detect and classify business stores based on their signboards. The new generation of object detectors, AKA one-stage object detectors as elaborated in Chapter 2, is able to detect and classify the objects at the same time. In this research, we tried to utilize such algorithms to detect and classify shop signboards. We had one object, which is the store signboard, and six shop classes defined in Chapter 4 and illustrated in Figure 4.5. The most obvious way to differentiate classes from each other using their signboards is the text within the signboards. As claimed in [13], their deep learning model has learned to utilize the text when needed for the purpose of multi-labeling stores. Therefore we did our pilot test, which is detailed in section 5.1, on the ShoS dataset in order to detect the signboard and classify the shop simultaneously applying one-stage deep learning algorithms. However, our results were unexpected, as the used model correctly detected the signboards but had poor store classification performance. Hence we built our Shop Signboard Detection and Classification Framework SSDCF with multiple stages in order to achieve our final goal.

The proposed SSDCF framework is composed of three main components to provide an end-to-end system that takes a raw image of a street view containing single or multiple storefronts and classifies it as follows. First, the detector locates the signboard from the input street view image. Then, the output bounding boxes of the potential signboards with high confidence scores are fed to an optical character recognition OCR component which extracts the text out of the signboards. Finally, the extracted text is processed through a natural language processing NLP classifier in order to classify the shop. Figure 5.1 shows the abstract illustration of the shop signboard detection and classification framework SSDCF. This required training the detection and the classification models in addition to choosing one of the state-of-art OCRs and evaluating it in order to produce good overall performance.

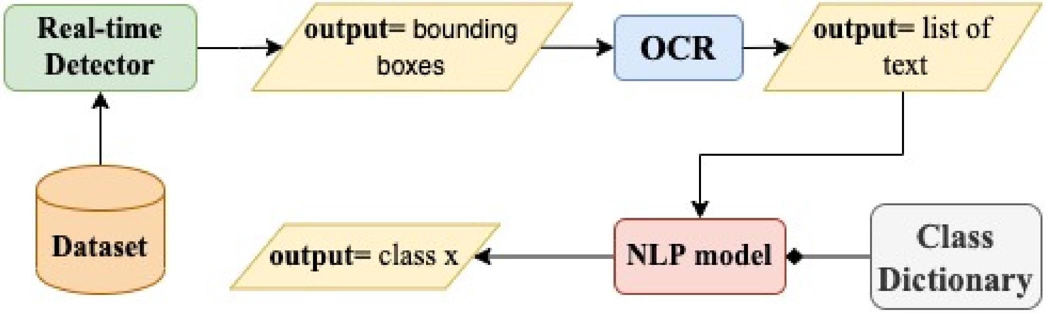


Figure 5.1: The abstract view of the proposed end-to-end SSDCF

In this chapter, the pilot study will be explained and discussed in Section 5.1. Then, the experiments that were conducted for shop signboards detection are going to be elaborated in Section 5.2 followed by text extraction trials in Section 5.3. Next, store classification using NLP techniques will be detailed in Section 5.4.

5.1 Pilot Study

In the pilot test, we used YOLOv3 [6] object detector to detect store signboards and classify them at the same time. The ShoS dataset was pre-processed as mentioned below in section 5.2.1 except that the number of classes C was set to six shop classes. Also, splitting the data into training and testing sets was performed differently by considering two factors: 1) the balance between single stores per image versus multi-stores per image, and 2) the balance among all six classes.

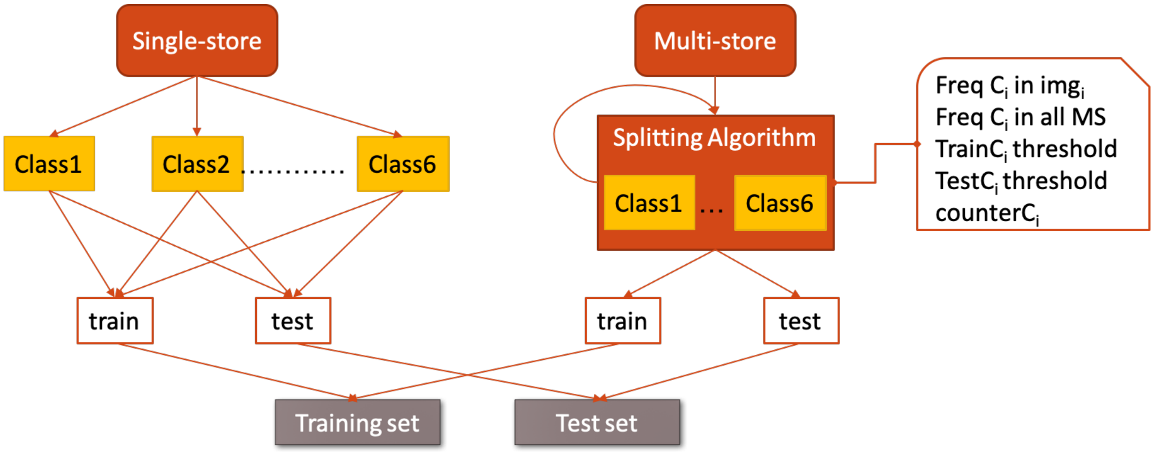


Figure 5.2: The splitting steps of the ShoS data into the training and testing sets for the pilot test

In order to meet both requirements, the image set was split into two groups: single stores per image and multi-stores per image. For the first group, a proportionally equal number from each class was taken into the training and testing sets after randomly shuffling them while maintaining the 80/20 ratio. For the second group, which has multi-stores per image, the frequency of occurrence of each class was identified for each image, and the frequency of each class was identified across the whole group. Then, a recursive loop was implemented for all classes starting with the least frequent class. The loop selects images for the training group while ensuring the image contains at least one signboard from the least frequent class. Once an image is selected for the training group, the counters for all classes are updated and the loop runs again. The process ends for each class if the 80% threshold for that class is reached. The same process is repeated for the testing set. Figure 5.2 illustrates the steps for splitting the ShoS images into the training and testing sets.

The training was performed on NVIDIA Geoforce RTX 2070 GPU with 16 GB of RAM utilizing the Darknet framework. The training was stopped after 50k iterations. The results were not satisfactory as it produced a mAP@0.5 =29.02% and a recall=38% with a confidence score equals to 0.25. Table 5.1 shows the average precision AP for each store class where the highest AP was registered for Finance and Investing class with 48.71% only. The outcomes were investigated by visualizing them on the validation set. It was noticed that the model was able to detect signboards

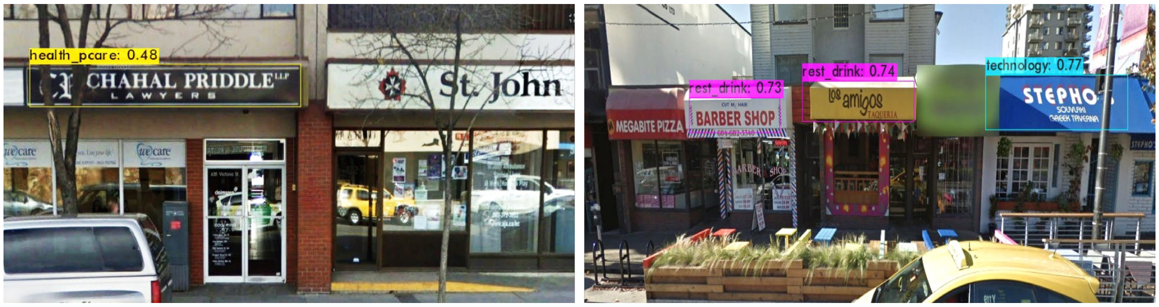


Figure 5.3: The output of some testing sample images from the pilot study for signboard detection and store classification

Table 5.1: The dissatisfying results of the pilot study for all classes

Class	**AP%**	**Class**	**AP%**
rest_drink	32.18	finance_investing	48.71
food_beverage	35.66	fashion	15.62
health_pcare	33.10	technology	22.21

correctly but not classify them as shown in Figure 5.3. Therefore, we built up the framework illustrated in Figure 5.1 where store signboards will be detected first using a one-stage detector and then go to several phases to classify the store utilizing textual information.

5.2 Shop Signboard Detection

5.2.1 Experimental Setup

The ShoS dataset was pre-processed in order to train the detectors. All images were resized to 960x720 as this image size helped to balance the image quality and image processing time. Bigger image sizes could not be handled due to the huge amount of memory they required during training while smaller image sizes reduced the model performance. Then, two input resolutions were chosen for training: 640x640 and 320x320 plus two color schemes: RGB and grayscale to see how the resolution and color factors would affect the models' accuracy.

The data was split into training and test sets with a ratio of 80/20 taking into account the fact that some images have a single store while other images include multiple stores. Hence, a proportionally equal amount from each group in the train and test sets were managed. Furthermore, about 11% signboards of the ShoS dataset, which represented the “Other” class, were blurred and provided as true negative samples.

The number of classes was set to one which is “Signboard”. Also, the number of batches was set as 64 and 16 for the smaller and bigger input resolutions respectively, and the learning rate was set to 0.001. Moreover, as suggested by [97], using image augmentation would make the model robust to different variations. So, some techniques were used, such as shift and zoom with respect to the overall structure of the image is not destroyed.

5.2.2 YOLO and SSD Detectors

Based on the literature of detecting licence plates [66, 68, 69, 70] and road traffic signs [25], real-time detectors showed robust performance. Therefore, one-stage object detectors YOLOv3 [6] and SSD [9] were utilized to detect shop signboards as they stand out among other available models in terms of accuracy and speed. The models make several predictions in a single pass using convolutional neural networks. Several backbones were used based on the object detectors where the YOLOv3 model used Darknet53 pre-trained on the ImageNet dataset [35] and the SSD model used MobileNetv2 FPNLite pre-trained on COCO dataset [36]. Applying transfer learning helps to boost performance by leveraging the knowledge gained from previous learning.

The training was stopped after 5k iterations for YOLO and after 8k iterations for SSD as the results were plateauing. The output of the inference stage was a bounding box for each signboard coupled with its confidence score. The confidence score C indicates how sure and accurate the model is regarding the bounding box containing a signboard. It was computed for each predicted signboard using the Jaccard index (AKA Intersection over Union IoU) and class probability using the Formula 10. Figure 5.4 shows an illustrated sample for the signboard detector output.

$$C = P(object) * IoU \tag{10}$$

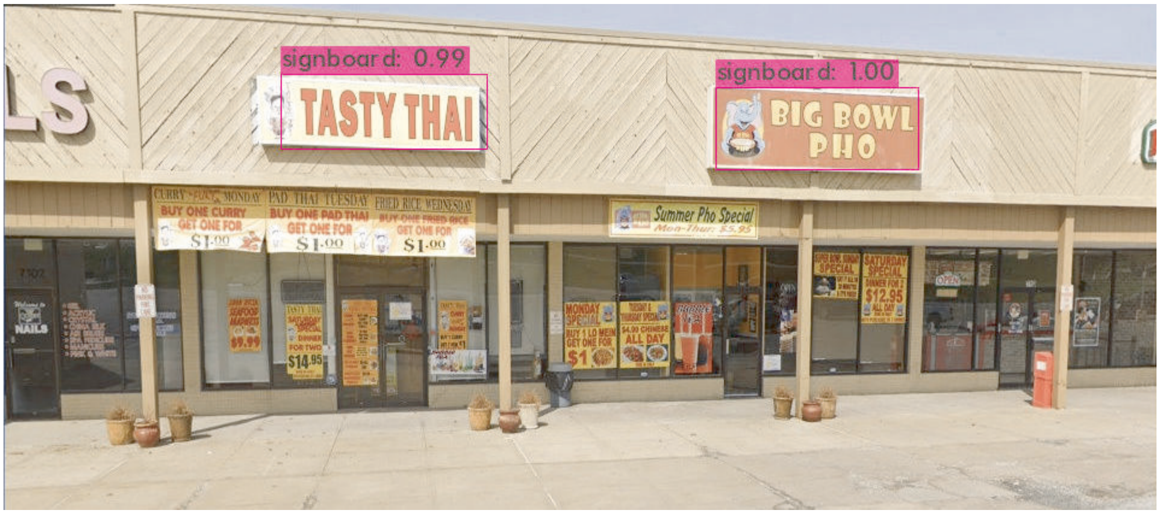


Figure 5.4: An illustrated sample for the signboard detector output

5.3 Text Extraction

To extract text from the detected signboards, two Optical Character Recognition OCRs were selected: ABBYFineReader [44] and Google Cloud Vision OCR [45] as they stand out among the others reviewed OCRs in the literature in Section 3.4. ABBYY FineReader OCR is a state-of-art commercial OCR [98, 99], and it was supplied by CENPARMI through a licensing agreement. Although its algorithm is not disclosed to the research community, it is claimed that it is AI-based. Also, it is capable of analyzing the overall layout to define the text areas for recognition along with format information. It accepts a wide range of file formats for input and output, pre-processes images to enhance their quality, and recognizes text of different styles and languages.

On the other hand, Google Cloud Vision OCR is an open source Application Programming Interface API that enables text detection and recognition from digital images with support for multiple languages. Developers can get their own API key to integrate Google Cloud Vision into their application including mobile applications. Its performance is remarkable in general, and it achieved better results when compared to Microsoft Cognitive Services OCR tool [83]. The quality attributes for both OCRs are described in Table 5.2.

Table 5.2: A comparative quality attributes for the OCRs used in this research

Quality Attributes	**ABBY FineReader**	**Google Cloud Vision OCR**
Open Source	No	Yes
Cloud Service	Yes	Yes
Multilanguage Support	Yes (up to 210 languages)	Yes (up to 60+ active languages*)
OS Support	Any	Windows, Mac, Android
AI-based	Yes	Yes
Free	No	Yes (partially)
Supported Input Format	text, table, presentation, image formats, and PDF	JPEG, GIF, PNG, TIFF, BMP, WEBP, and PDF
Supported Output Format	TXT, XML, PDf, CSV, RTF, HTML	TXT, JSON, CSV

**Active language means they are prioritized and regularly evaluated by Google*

ABBY FineReader OCR build 15.0.115 and Google Cloud Vision OCR v1 were fed with the ShoS-cropped dataset, which contains 10k signboard images, to extract text out of the shop signboards. The OCRs produced a text file for each image containing all extracted text data as illustrated in Figure 5.5.

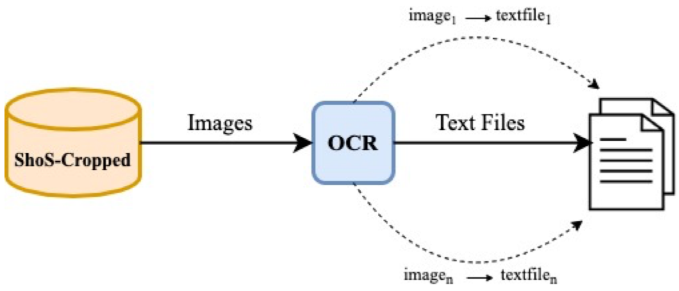

$\text{image}_1 \rightarrow \text{textfile}_1$
ShoS-Cropped
Images
OCR
Text Files
$\text{image}_n \rightarrow \text{textfile}_n$

Figure 5.5: The process of extracting text from the ShoS-Cropped images using OCR

5.3.1 Post-processing

Prior to the evaluation process, some cleaning steps were implemented for both ground truth and OCR text files. First, all text data was converted to lower case. Then, all non-Latin, non-alphabetic, numbers, and empty items were removed to avoid unnecessary comparisons. Thus, two clean sets for the ground truth and the OCR extracted text were ready for evaluation which is detailed in the next chapter.

5.4 Shop Classification

In the classification stage, the textual information that appeared on shops' signboards was utilized. To train the models, the ShoS annotation file was used. From the 10k annotated signboards, two attributes were used for the text classification process: the text data on the signboards and the associated shop class. There are six shop classes in the ShoS dataset as elaborated in Chapter 4. The distribution of all store classes in the ShoS dataset was already illustrated in Figure 4.6. Multiple phases were implemented to classify shops based on the textual information that appears on their signboards: 1) text cleaning, 2) feature extraction, and 3) training/testing the models. Figure 5.6 shows the phases of store classification. In the following discussions, each signboard text sample is called a document.

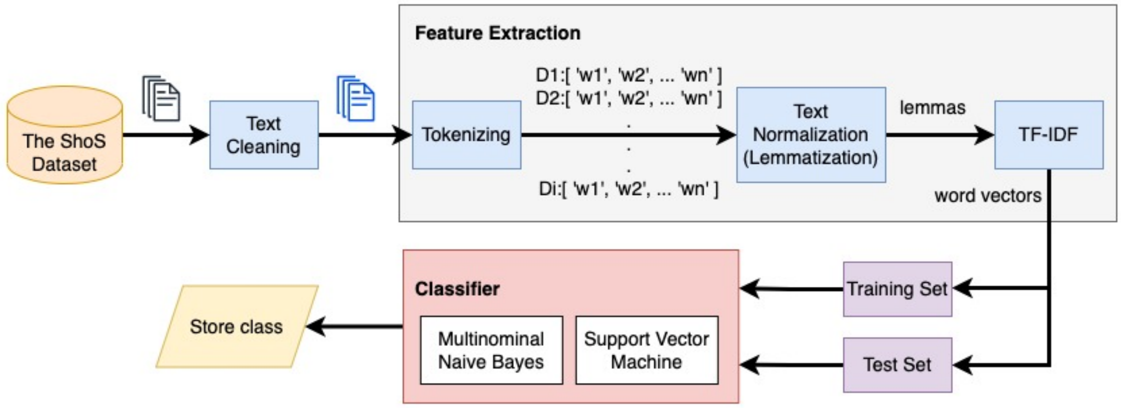


Figure 5.6: The phases of store classification based on the textual information

5.4.1 Text Cleaning

Since documents can contain some sources of noise, the following cleaning steps were performed to enhance the quality of the text data. The document size was reduced to 8904 documents after cleaning.

(1) Convert the text data to lowercase.

(2) Remove superfluous text data like punctuation and words with two or fewer characters.

(3) Remove stop-words, which are words that do not add meaning like “the, in, are ...” using Natural Language Toolkit NLTK.

(4) Remove numbers from the text data as they are insignificant for the classification process.

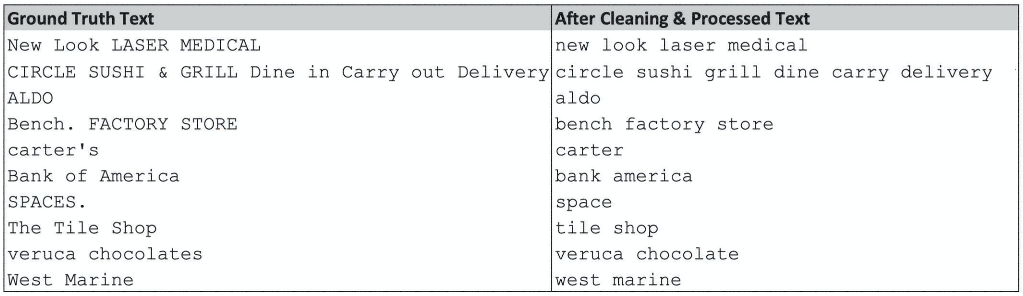
(5) No spelling correction was needed as text data in signboards does not necessarily follow correct spelling.

(6) All duplicate documents, which is possible because of the existence of chain stores and other common naming, were removed.

5.4.2 Feature Extraction

The feature extraction process started with tokenizing all words in each document in the ShoS dataset. Then, a text normalization technique was applied to prepare the text data for the classification process. Stemming and Lemmatization approaches are usually used with NLP problems in order to reduce morphological variations of words. The difference between them is that stemming removes the last few characters of a word and usually results in incorrect meaning and spelling, which is called Stem. On the other hand, Lemmatization considers context by converting the word to its meaningful base form, which is called lemma. For instance, the words {changing, changed, and change} have the stem “chang” and the lemma “change”. In this research, the Lemmatization technique is applied as its accuracy is paramount and the ShoS dataset is not huge compared to NLP problems. Figure 5.7 shows a comparison of text data prior to and after cleaning and pre-processing.

Since each document in the corpus of the ShoS dataset belongs to one class only, we wanted to quantify words and assign a weight to each word in order to keep the focus on significant keywords that carry a value for each store class. To implement that, the Term Frequency-Inverse Document Frequency TF-IDF technique was used. TF-IDF vectorizes all the text data at a word-level in order



Ground Truth Text	After Cleaning & Processed Text
New Look LASER MEDICAL	new look laser medical
CIRCLE SUSHI & GRILL Dine in Carry out Delivery	circle sushi grill dine carry delivery
ALDO	aldo
Bench. FACTORY STORE	bench factory store
carter's	carter
Bank of America	bank america
SPACES.	space
The Tile Shop	tile shop
veruca chocolates	veruca chocolate
West Marine	west marine

Figure 5.7: A comparison of text data prior to and after cleaning and feature extraction in text classification stage

to classify the documents. The word vector is computed using the following equations 11, 12, 13, 14, where t is the word, d is the document (set of words per signboard), N is the count of corpus, and $corpus$ is the total document set.

$$TF\text{-}IDF(t,d) = TF(t,d) \times IDF(t) \tag{11}$$

$$TF(t,d) = \frac{count\ of\ t\ in\ d}{number\ of\ words\ in\ d} \tag{12}$$

$$DF(t) = occurrence\ of\ t\ in\ documents \tag{13}$$

$$IDF(t) = log(\frac{N}{DF+1}) \tag{14}$$

5.4.3 MNB and SVM Classifiers

Two classical classifiers were chosen, Multinomial Naive Bayes MNB and Support Vector Machine SVM, over deep learning methods. The decision was based on the following reasons: 1) text classification based on deep learning algorithms requires a tremendous amount of data for training compared with traditional methods [47]. The size of our corpus (i.e. the number of text documents in the ShoS dataset for shop classification) is relatively small; 2) the computational cost for deep learning approaches might get complicated during training and, 3) MNB and SVM stand out among the other reviewed classifiers [85, 86, 87] with robust performance for text classification problems, such as topics, news, and emails classification.

The vectorized data was split into training and testing sets with 70/30 ratio. The training set was fed into MNB and SVM classifiers. The “Other” class was excluded to avoid confusing the classifiers as samples of that class do not share common features. For the hyper parameters for MNB alpha=1.0, class_prior=None, and fit_prior=True; while in SVM, the weight of all classes was set by default to one, RBF kernel was used, squared hinge for loss function with 1000 max iteration and a penalty equal to 12 with no random state. The final result is a store class, and it is considered correct if it matched the ground truth.

To enhance the classifiers performance, we re-trained the models with the ground truth data plus augmented data. Augmentation is a common technique in the computer vision field and its effectiveness has been proven in NLP problems too [100, 101]. There are various text augmentation approaches such as paraphrasing-based, noising-based, and sampling-based [100]. These approaches can be utilized based on the objective of the augmentation and the level of text analysis i.e. phrase, sentence, or word. Wie and Kai have demonstrated in [102] that the labels of the augmented text are most likely to be the same as on the labels of the original text. Thus, there is no need to re-label the augmented documents. Nosing-based techniques, AKA Easy Data Augmentation EDA, were selected for the purpose of this research as they are better suited for word-level problems. Particularly, Random Deletion RD and Thesauruses-inspired methods were selected with some adjustments detailed below.

In the Random Deletion RD method, n number of words are randomly deleted from the training samples with a specified threshold α. Since each document of our corpus contains a limited number of words ranging between 1 and 32, we lowered the threshold to $\alpha = 0.2$ and computed n with the Formula 15 where l is the number of words per document. That means if we have a document with 10 words, only two random words would be deleted. The maximum number of words to be deleted was six.

$$n = \alpha \times l \tag{15}$$

In the Thesauruses method, words in the original text are replaced with true synonyms and hyponyms [100, 103]. Since our work is based on the extracted text by the OCR, we came up with a thesauruses-inspired method, named *OCR-Thesauruses*. The objective of this method is to generate augmented text by replacing some words from the true documents with similar misrecognized words, named *miz* words, from the generated *OCR-Thesauruses*. Then, the augmented data will be added to the training set in order to improve the classifiers' performance. To do that, we first generated the *OCR-Thesauruses* which is a set of *miz* words for each class. Then, it was used to generate the augmented text documents.

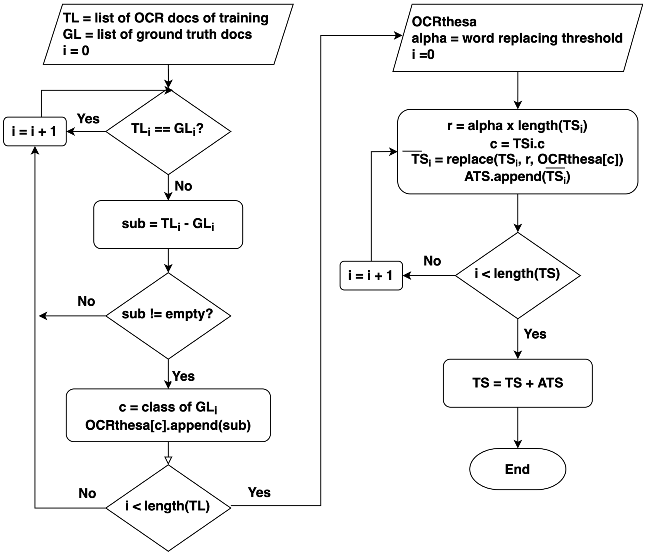

TL = list of OCR docs of training
GL = list of ground truth docs
i = 0
Yes
i = i + 1
TL$_i$ == GL$_i$?
No
sub = TL$_i$ - GL$_i$
No
sub != empty?
Yes
c = class of GL$_i$
OCRthesa[c].append(sub)
No
i < length(TL)
Yes
OCRthesa
alpha = word replacing threshold
i =0
r = alpha x length(TS$_i$)
c = TSi.c
$\overline{TS_i}$ = replace(TS$_i$, r, OCRthesa[c])
ATS.append($\overline{TS_i}$)
No
i = i + 1
i < length(TS)
Yes
TS = TS + ATS
End

Figure 5.8: A flow chart for OCR-Thesauruses and ATS generation

In order to explain the *OCR-Thesauruses* method, let us assume that TL refers to the list of documents that were extracted by the OCR, GL refers to the list of documents of the ground truth text of the ShoS dataset, and C is a set of six shop classes. Each instance in the TL has a corresponding instance in the GL, and it is coupled with its class $c_k \in C$. Also, a matching score for each instance in the TL compared to its true instance is known based on previous evaluation. The *OCR-Thesauruses* is created as follows: for each document TL_i in TL, the matching score was checked where i is the number of documents in TL. If an exact match was captured, the document was discarded to avoid duplicates. Then, we subtracted GL_i from TL_i to get the difference which is a set of words that are in TL_i but not in GL_i. If the subtraction set was empty, which means that there is a deletion only and no insertion or substitution to the true document, the document was discarded as it does not provide us with *miz* words. Otherwise, a set of *miz* words for each store class C was generated producing $OCRthesa$. The $OCRthesa$ is a map with C keys and a set of *miz* words assigned for each key. Finally, the augmentation was done by replacing some words from each document in the original training set TS with words that have high similarity from the corresponding class of *OCR-Thesauruses*. The similarity was computed using the Python library difflib.SequenceMatcher [1]. This way we ensured that words that are more similar to the true words have a higher probability of being chosen. The number of replaced words considered the size of the document and computed based on a specified threshold set to 0.5. The whole process is illustrated in Figure 5.8 and provided in pseudo code in Algorithm 1.

Implementing the above text augmentation methods utilizing RD and *OCR-Thesauruses* allowed adding about 7k augmented documents to the training set. So, both classifiers MNB and SVM were retrained using the new training set following the same procedure of the previous experiment. The results are shown in Chapter 6, and it is mentioned to this experiment by Augmented Training Experiment ATE.

[1] https://docs.python.org/3/library/difflib.html

Algorithm 1: OCR-Thesauruses and ATS generation

```
Input: $TS$: training set
$TL$: list of OCR documents of TS
$GL$: list of ground truth document
$C$: set of 6 store classes
$\alpha$: word replacing threshold
Output: ATS: augmented training set
$i \leftarrow 0$
for $TL_i$ in $TL$ do
    if $TL_i \neq GL_i$ then
        $sub \leftarrow \emptyset$
        $sub \leftarrow TL_i - GL_i$
        if $sub \neq \emptyset$ then
            $c \leftarrow 0$
            $c \leftarrow GL_i.class$
            $OCRthesa[c].append(sub)$
        end
    end
    $i \leftarrow i + 1$
end
$i \leftarrow 0$
for $TS_i$ in $TS$ do
    $l, r, c \leftarrow 0$
    $l \leftarrow length(TS_i)$
    $r \leftarrow \alpha l$
    $c \leftarrow TS_i.c$
    $\overline{TS_i} \leftarrow replace(TS_i, r, OCRthesa[c])$
    $ATS.append(\overline{TS_i})$
end
$TS \leftarrow TS + ATS$
```

Chapter 6

Performance Measures

This chapter shows the evaluation methods for the experiments conducted in the previous chapter. The results are revealed and discussed for store signboard detection in Section 6.1, text extraction in Section 6.2, and store classification in Section 6.3. Also, the full framework evaluation is presented and discussed in Section 6.5. Finally, the classifier performance is compared to human performance in Section 6.4.

6.1 Shop Signboard Detection

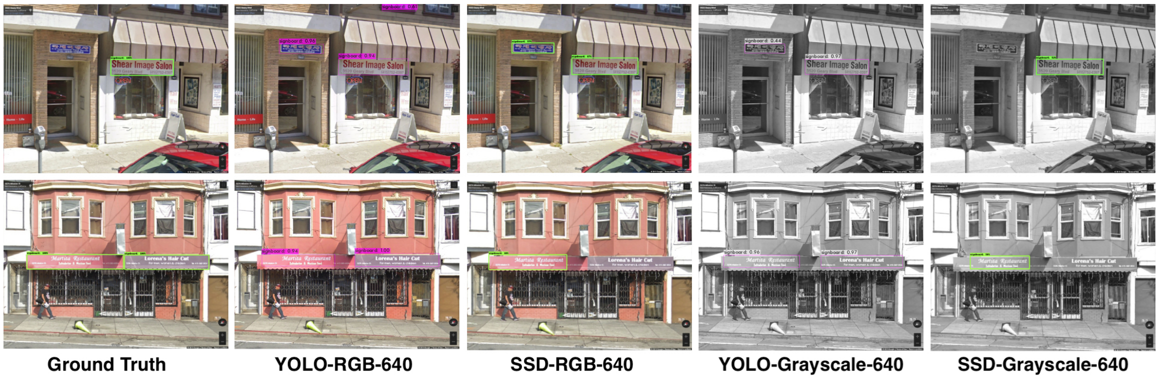
The experiments were performed for both models YOLO and SSD on Google Colab Pro[1] which assigned a GPU machine with an option of "high-RAM" usage. Detection was considered correct if the value of intersection over union IoU was 0.5 or higher as recommended in previous studies of license plate detection [66].

The models were tested for several variations of average precision over the four options mentioned before for the input resolutions and color schemes. Based on the results shown in Table 6.1, YOLOv3 with an input resolution of 640x640 and an RGB color scheme produced the best results reaching a mean average precision of 94.23% at IoU=0.5 with a confidence score set to 0.25. Therefore, the spatial distribution of labels directly impacts the detection of small objects (signboards) at

[1] https://colab.research.google.com

Table 6.1: Results of one-stage detectors over mean Average Precision of 0.5 and 0.75 and the recall at IoU=0.5

Detector	Input Resolution	Image Color	mAP@0.5(%)	mAP@0.75(%)	Recall%
YOLO3	320x320	RGB	92.0	66.09	91
	640x640	RGB	**94.23**	**76.76**	**94**
	320x320	Grayscale	91.2	64.09	89
	640x640	Grayscale	91.88	69.0	91
SSD-Moblilenet2	320x320	RGB	88.8	74.08	54
	640x640	RGB	90.4	76.84	53
	320x320	Grayscale	85.5	70.0	49
	640x640	Grayscale	84.6	68.8	47



Ground Truth YOLO-RGB-640 SSD-RGB-640 YOLO-Grayscale-640 SSD-Grayscale-640

Figure 6.1: Results of store signboard detection from YOLOv3-640 and SSD-Mobilenetv2-640 in two color schemes compared to the ground truth

lower resolutions as they are harder to detect. Figure 6.1 displays some sample results obtained by YOLOv3-640 and SSD-Mobilenet2-fpnlite-640 models which demonstrates the superiority of YOLOv3. The YOLOv3 model was robust enough to predict even partially occluded signboards. It is also noticed that both detectors could detect most of the signboards, even the ones that were missed by the annotators intentionally because of the limitations mentioned in Section 4.5. Figures 6.2 and 6.3 illustrate the performance of both methods in terms of mean average precision and average loss. It is noticed that the average loss for all YOLOv3 variations reached a value less than 1 (0.07 in its ultimate case). On the other hand, the loss was computed differently in SSD because the final loss is a combination of cross-entropy loss and Smooth L1 loss. It reached a small value too (0.02 in its ultimate case).

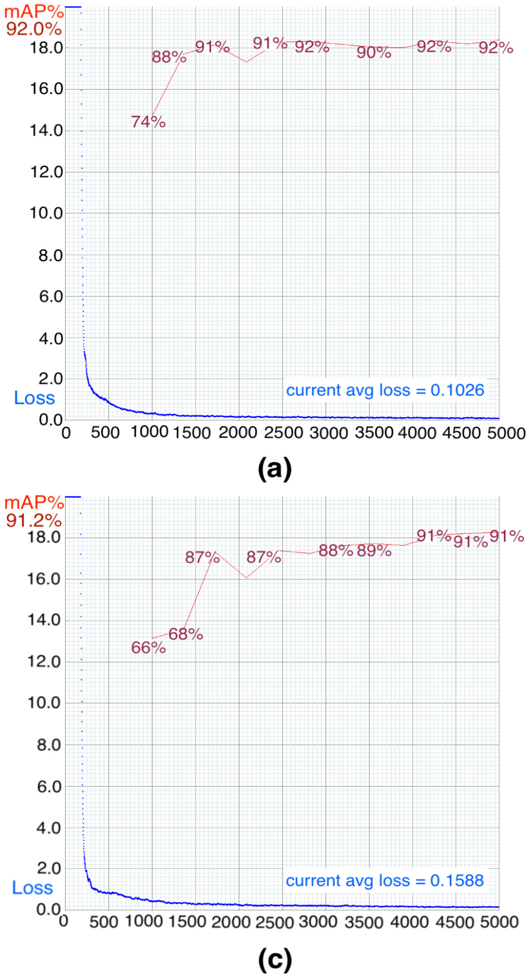


Figure 6.2: Average loss (blue) and mean average precision mAP (red) for YOLO experiments. (a) RGB-320; (b) RGB-640; (c) Grayscale-320; (d) Grayscale-640.

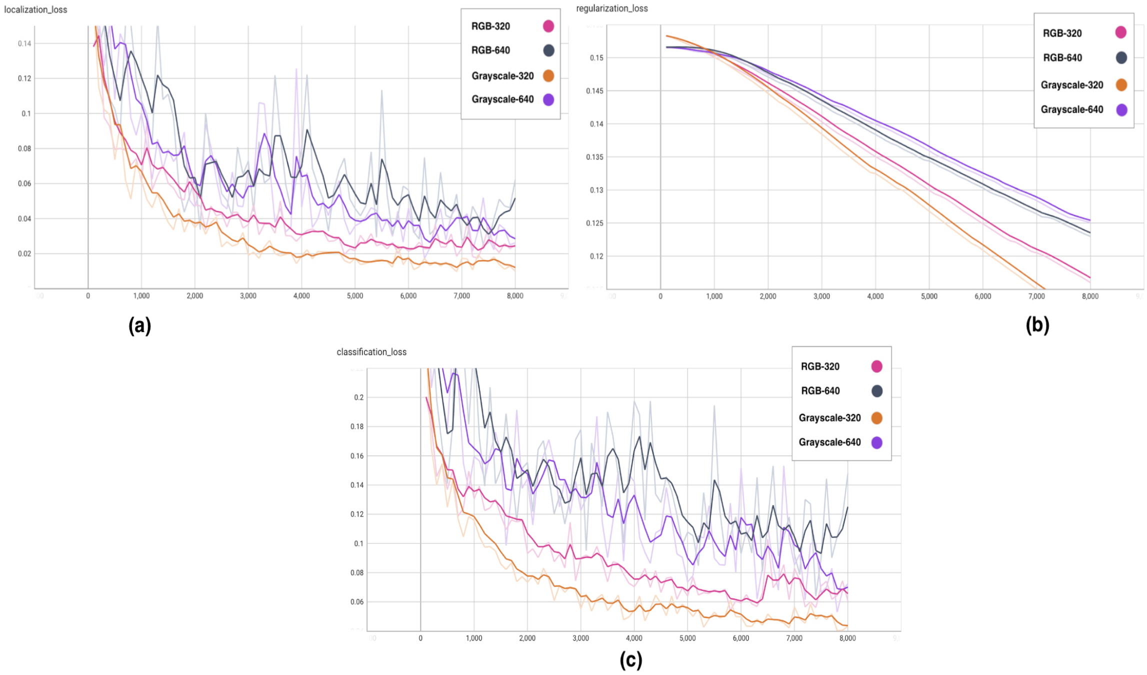


Figure 6.3: The loss charts of: (a) Localization, (b) Regularization, and (c) Classification for SSD experiments with a smooth factor set to 0.6

Finally, it was observed that when we provided the detectors with true negative samples i.e. blurred signboards, results were significantly higher by 10% approximately in terms of mean average precision. That negates the assumption made for one-stage detectors which ignores training the models with true negative samples. That was not the case at least in our research area.

6.2 Text Extraction

To evaluate ABBYFineReader and Google Cloud vision OCR, the Levenshtein Distance LD string metric [46] was used at first. LD compares two strings: OCR text s and the ShoS-cropped ground truth text t. The distance score is calculated based on the number of deletions, insertions, and substitutions needed to transform s into t using Equation 16 where i is the terminal character position of t, and j is the terminal character position of s. The lower the scores are the better, and zero is a perfect match. The average score for ABBYFineReader OCR, Google Cloud Vision OCR, and the percentage of total exact matching are presented in Table 6.2. The performance of Google

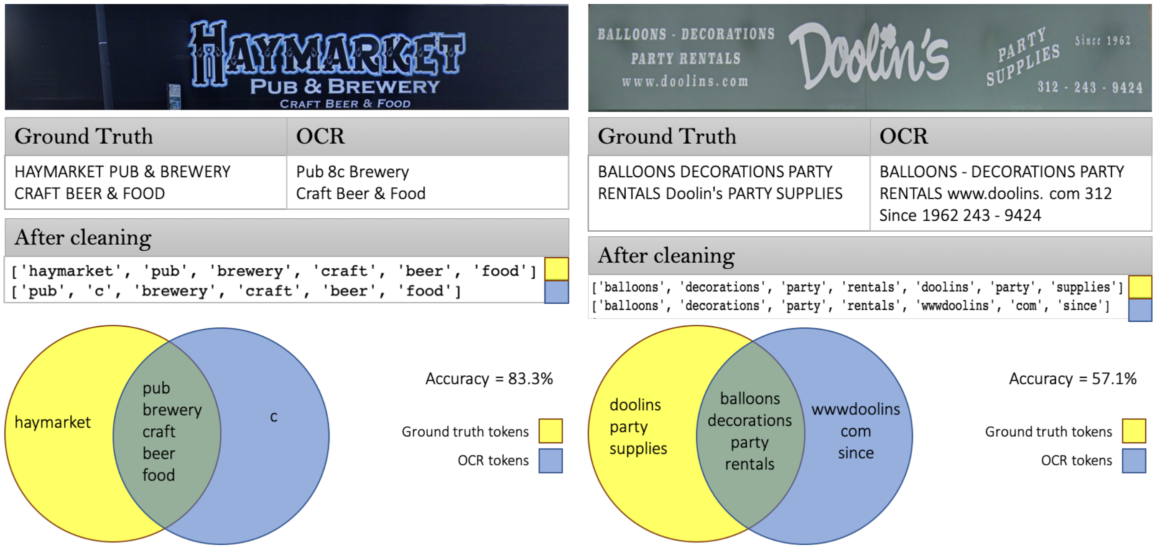
Cloud Vision OCR was better than ABBYFineReader as the latter had some difficulties with stylized typefaces. Thus, Google Cloud Vision OCR is preferred over the other one.

$$lev_{t,s}(i,j) = \begin{cases} max(i,j) & if min(i,j)=0, \\ min \begin{cases} lev_{t,s}(i-1,j)+1 \\ lev_{t,s}(i,j-1)+1 \\ lev_{t,s}(i-1,j-1)+1_{(t_i \neq s_j)} \end{cases} & otherwise \end{cases} \quad (16)$$

Table 6.2: The Levenshtein distance (LD) average score with the percentage of total exact matching in the prior evaluation method in addition to the IAA evaluation method under two thresholds of the used OCRs

	LD average score	**Exact matching**	**IAA accuracy (th=100%)**	**IAA accuracy (th=50%)**
ABBYFineReader	12.37	22.07%	49.1%	83.7 %
Google Cloud Vision	**7.02**	**42.54%**	**82.4%**	**89.7%**

Unlike document analysis research, our work paid less attention to the exact matching of the extracted text by the OCR compared to the ground truth because of two main factors: 1) the annotation of the ShoS-cropped dataset ignores some text on signboards such as numbers, addresses, and non-English characters, and 2) this research focuses only on keywords that play an important role in the classification stage. Therefore, we re-evaluated the accuracy of OCR as follows: for each signboard image, two lists were created: ground truth list gt_list and OCR list ocr_list by tokenizing the words. Then, both lists were cleaned by removing all non-alphabetical characters and numbers and converting them into lower case. Next, a new result list $result\_list$ was generated by intersecting gt_list and ocr_list (Equation 17). The accuracy was computed by dividing the length of $result\_list$ by gt_list (Equation 18). The intersection average accuracy IAA was calculated for all images using Equation 19, where n is the number of signboard images. The intersection average accuracy was computed using two matching thresholds: 100% and 50%. Figure 6.4 illustrates two examples of IAA evaluation with a matching threshold equals to 100. The overall intersection average accuracy for both OCRs is presented in Table 6.2 where Google Cloud Vision performs better than ABBY FineReader in this evaluation methodology too. That might be attributed to the huge



Ground Truth	OCR
HAYMARKET PUB & BREWERY CRAFT BEER & FOOD	Pub 8c Brewery Craft Beer & Food

After cleaning

['haymarket', 'pub', 'brewery', 'craft', 'beer', 'food'] 🟨
['pub', 'c', 'brewery', 'craft', 'beer', 'food'] 🟦

Ground Truth	OCR
BALLOONS DECORATIONS PARTY RENTALS Doolin's PARTY SUPPLIES	BALLOONS - DECORATIONS PARTY RENTALS www.doolins. com 312 Since 1962 243 - 9424

After cleaning

['balloons', 'decorations', 'party', 'rentals', 'doolins', 'party', 'supplies'] 🟨
['balloons', 'decorations', 'party', 'rentals', 'wwwdoolins', 'com', 'since'] 🟦

Figure 6.4: Elaborating examples of the IAA evaluation method for the tested OCR with a 100% matching threshold

amount of data that Google has used for building and training its OCR.

$$result\_list = gt\_list \cap ocr\_list \tag{17}$$

$$accuracy = \frac{length(result\_list)}{length(gt\_list)} \tag{18}$$

$$IAA = \frac{\sum_{i=1}^{n} accuracy_i}{n} \tag{19}$$

6.3 Shop Classification

The results of testing MNB and SVM classifiers to classify shops based on the text appeared on their signboards reached an accuracy of 85.74% and 90.01% respectively. When augmented data were added to the training set in the ATE, accuracy had been increased by about 4% where MNV accuracy became 88.97% and SVM accuracy became 94.11%. Table 6.3 shows the precision, recall, and f1-score for all classes for both classifiers before and after adding the augmented data in the ATE. Since F1-score is the most preferable measure for such problems, we illustrated in Figure 6.5 a comparative bar chart for each class in both classifiers with and without data augmentation. It is noticed that SVM works better even with classes that have less samples and more non-descriptive text like class “Fashion” and “Technology”. By looking at F1-scores of the SVM, it is observed that class “Fashion” has the lowest performance and that might be because of the high possibility of non-descriptive names used in their signboards as it represents 30% of the non-descriptive samples. In contrast, the class “Finance and Investing” has stronger performance because of the consistency and limitations of the vocabularies that could be used in their signboards. Figure 6.6 illustrates the confusion matrices for both classifiers, and the increase in performance are obvious for all classes. The macro F1-scores for both classifiers are also shown in Figure 6.7 and Table 6.4. When manual verification was performed on some samples of the confusing cases, it was observed that the confusion was caused by the usage of misleading words that are unrelated to store class such as “Nail Bar”, or because of the common vocabularies between two classes like the word “food” in “rest_drink” and “food_beverage” classes.

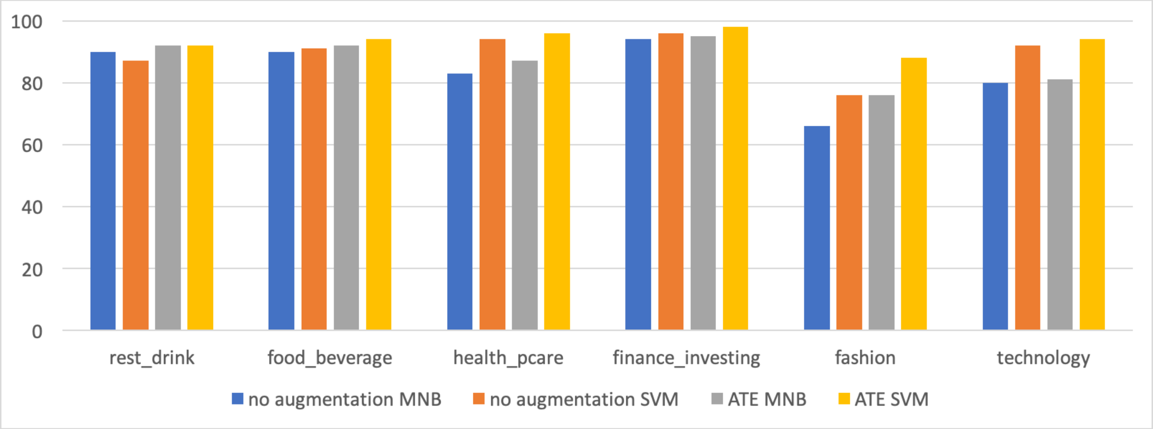


Figure 6.5: The F1-score results of MNB and SVM classifiers with no augmentation and with augmentation training ATE

Class	Evaluation metrics	No augmentation		ATE	
		MNB	SVM	MNB	SVM
rest_drink	Precision	91%	80%	92%	87%
	Recall	88%	94%	92%	98%
	F1-score	90%	87%	92%	92%
food_beverage	Precision	92%	92%	94%	96%
	Recall	88%	90%	90%	92%
	F1-score	90%	91%	92%	94%
health_pcare	Precision	72%	96%	78%	97%
	Recall	97%	93%	98%	95%
	F1-score	83%	94%	87%	96%
finance_investing	Precision	92%	97%	93%	98%
	Recall	96%	96%	96%	97%
	F1-score	94%	96%	95%	98%
fashion	Precision	97%	87%	96%	93%
	Recall	50%	68%	63%	85%
	F1-score	66%	76%	76%	88%
technology	Precision	99%	95%	99%	99%
	Recall	67%	89%	68%	90%
	F1-score	80%	92%	81%	94%

Table 6.3: The results of the store classification stage for MNB and SVM classifiers for all the studied classes with no augmentation and with augmentation training ATE

6.4 Comparison with Human Performance

To further assess our classifier, the classification performance was compared with human performance using an online survey. An equal number of signboard text documents from each store class were randomly selected from the test set of the ShoS dataset. According to [104], long survey usually increases the prevalence of careless responding, where participants respond to survey questions without considering the content. Thus, we decided to provide a short survey with a total of 24 test samples. The samples included 50% difficult ones with non-descriptive text for each class. This insured that the survey had a similar level of difficulty to our classification experiment. In this survey, only text data were provided for participant as the target is to compare the results with our classifier which is based on text data only.

An online survey, built using Google Forms[2], was set to collect human responses on the samples

[2] https://www.google.ca/forms/about/

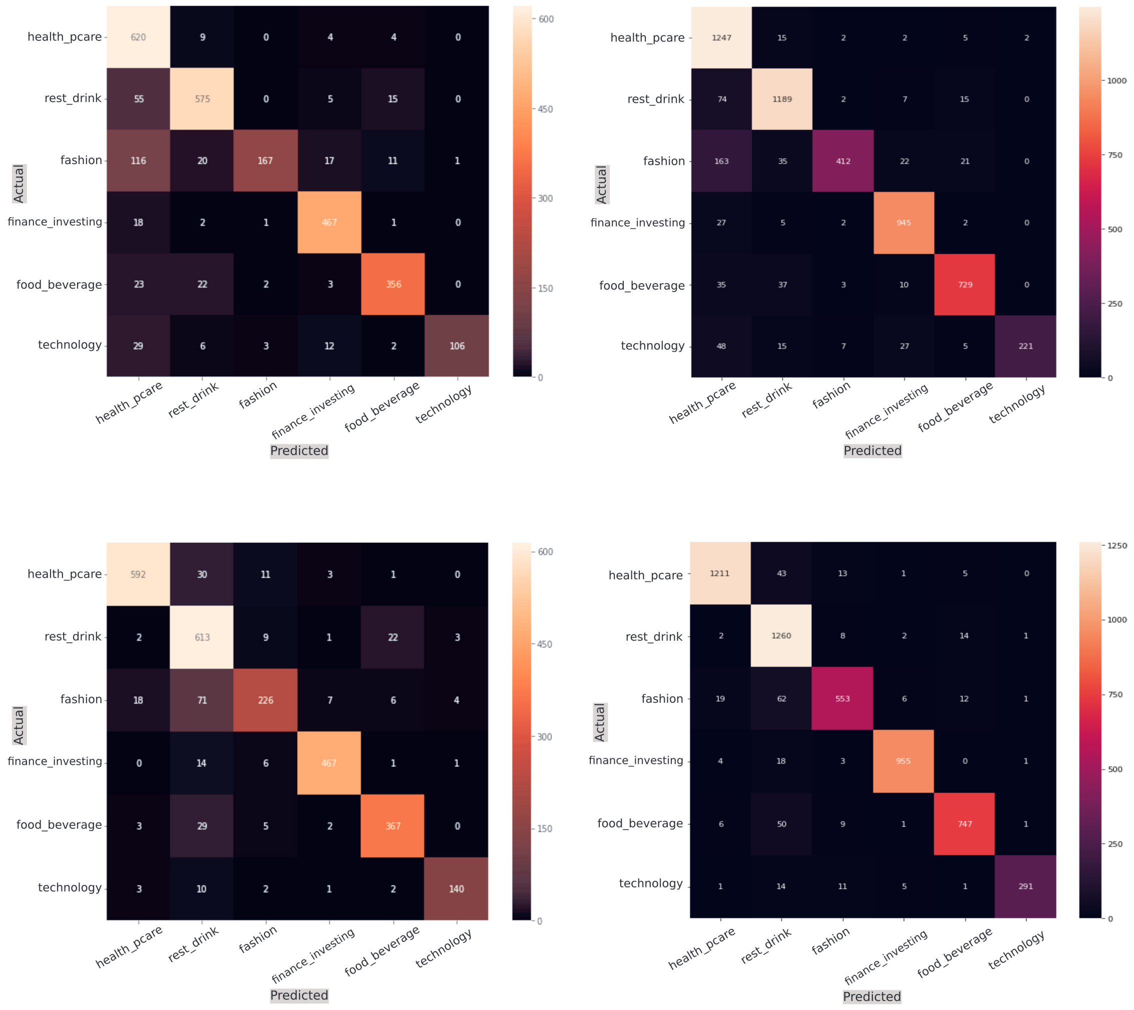


Figure 6.6: Confusion matrix heatmaps for MNB (up) and SVM (down) classifiers without augmentation (left) and with augmentation training ATE (right)

where each participant had to classify text samples based on the class tree they were supplied with. Figure 6.8 shows the class tree provided to the participants, and some of the test set documents along with a sample from the survey. At the beginning of the survey, the participants were provided with the purpose of the study in addition to simple instructions regarding how to classify the text. If the participant was not able to determine the designated class, he/she was guided to choose based on their best guess. The survey took about 10 minutes to complete.

Furthermore, the quality of the responses was assessed based on two factors derived from the

	MNB	SVM
No augmentation	83.83	**89.33**
ATE	87.17	**93.67**

Table 6.4: The macro F1-scores for MNB and SVM classifiers without augmentation and with augmentation training ATE

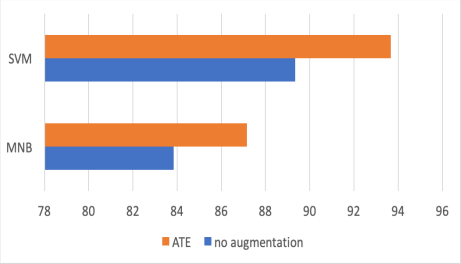
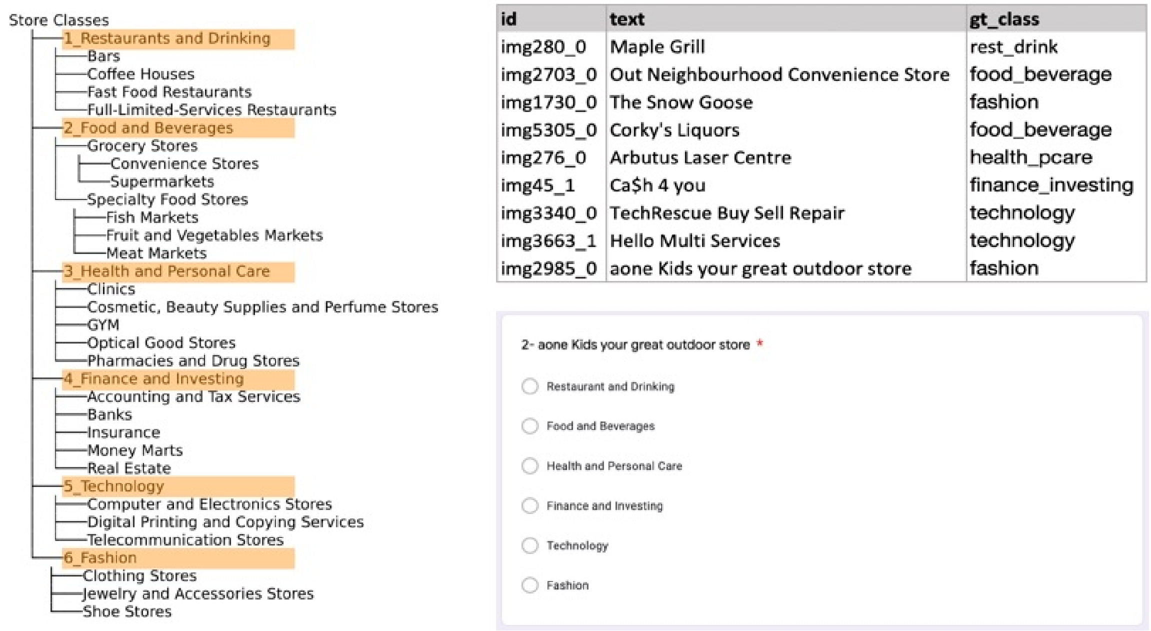


Figure 6.7: The Bar chart of macro F1-score for MNB and SVM classifiers without augmentation and with augmentation training ATE

collected personal information. The first factor was the participants' level of proficiency in the English language. All responses related to participants with an English proficiency level of beginner or lower were excluded. The second factor was the length of the participants' living experience in Canada and the US. All responses of participants who lived in Canada and the US less than 6 months were also excluded. This way we avoided any invalid assessments by eliminating the outliers.

The survey was distributed online through various communication applications. A total of 101 responses were collected. Females represented 52.4% and males represented 47.6% of the participant population. The survey results were analysed based on three measures: precision, recall, and F1-score. The results for all classes are included in Table 6.5. An illustration of the results for each store class is presented in Figure 6.9. Our classifier outperformed human performance by about 15% where it reached an F1-score=87.9% compared to 71.85% for human. Despite that the text annotation was done by human annotators, the results came positive to our classifier side. This is because of the knowledge that our classifier earned during training, which is based on pure text information clues. In comparison, when human subjects were provided with textual information only (i.e. no storefront or signboard images were provided), their classification was limited to the life knowledge they have in addition to the descriptive vocabularies within the signboard text document if existed! For example, the "Fashion" shop with this text on its signboard "The Snow Goose", could be difficult for human subjects to determine its class especially if he/she does not have any



id	text	gt_class
img280_0	Maple Grill	rest_drink
img2703_0	Out Neighbourhood Convenience Store	food_beverage
img1730_0	The Snow Goose	fashion
img5305_0	Corky's Liquors	food_beverage
img276_0	Arbutus Laser Centre	health_pcare
img45_1	Ca$h 4 you	finance_investing
img3340_0	TechRescue Buy Sell Repair	technology
img3663_1	Hello Multi Services	technology
img2985_0	aone Kids your great outdoor store	fashion

2- aone Kids your great outdoor store *

○ Restaurant and Drinking

○ Food and Beverages

○ Health and Personal Care

○ Finance and Investing

○ Technology

○ Fashion

Figure 6.8: The class tree provided to the participants in the human comparison survey (left), and Some of the test set documents along with a sample from the survey (right)

prior knowledge about the usage of goose feather in winter jackets. In contrast, if the model knowledge was built upon many occurrences of such vocabularies, it would be able to classify it correctly. This could be an interesting area to dig deeper and find solutions.

It was observed that the ambiguity factor for non-descriptive text resulted in the majority of the misclassified text by human participants. This highlights the importance of adding descriptive keywords related to store class in the signboards as it will improve the ability of both humans and machines to classify stores accurately especially when the store façade is not representative. Moreover, the only class human was able to achieve slightly better score than our classifier was "Restaurant and Drinking" and that was because of the common vocabularies between the mentioned class and "Food and Beverages" class. These similarities confused the classifier in most of the cases while human was able to get the semantic and differentiate them.

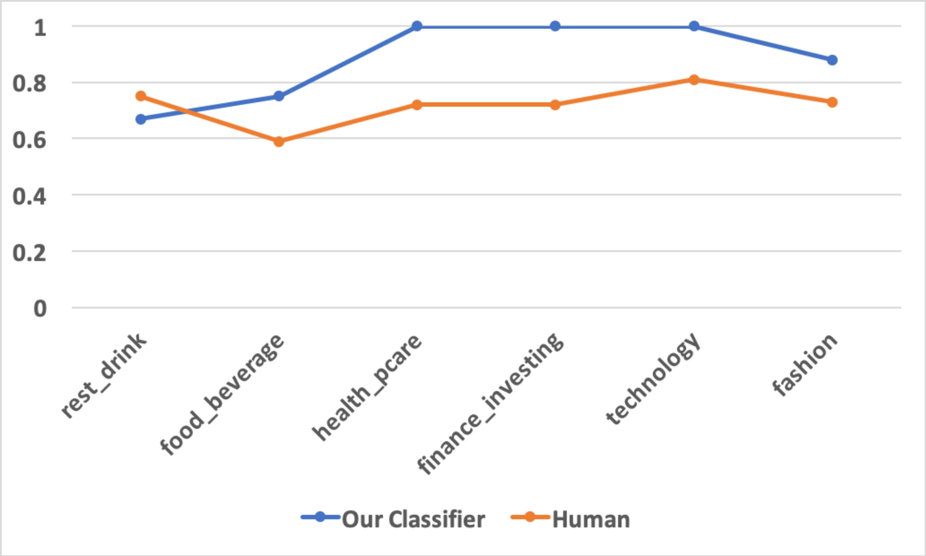


Figure 6.9: F1-scores for each store class for our classifier versus human

Table 6.5: The overall performance measures to compare our classifier versus human for all store classes

	Precision	**Recall**	**F1-Score**
Human	71.92%	71.97%	71.85%
Our Classifier	**89.17%**	**87.5%**	**87.9%**

6.5 SSDCF Evaluation

The three main components of our SSDCF framework: signboard detector, text extractor, and store classifier were evaluated using SVT dataset [10]. Sample images that contain storefronts of classes that are not included in this study, such as hotels and gas stations, were filtered. Because the ground truth annotation of the SVT dataset focuses on word locations within signboards, we regenerated the ground truth to fit our requirements. This was accomplished through the following steps: 1) all the bounding boxes of the shop signboards were automatically generated using our trained detector on the ShoS dataset, 2) the ground truth text of each detected signboard was recorded and reviewed manually, as the size of data is manageable, to ensure reliability, and 3) the ground truth of store class was also recorded manually in addition to other attributes like occlusion and difficulty. This way the SVT dataset was ready to be tested in our framework.

The street view images were fed to the best models according to the individual evaluations done

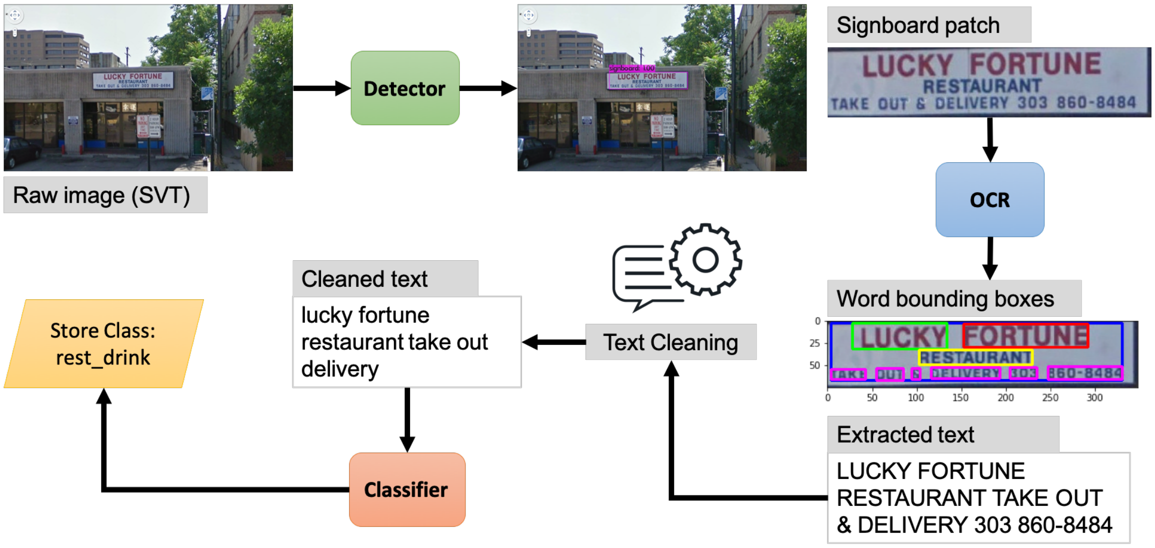


Figure 6.10: An illustration of the whole cycle of our SSDCF on a sample street view image from the SVT dataset

before for each component. First, the raw image of a street view was input into the YOLOv3 detector to detect shops' signboards. Then, the detected bounding boxes with confidence scores higher than 0.25 were pipe-lined into the Google Vision OCR to extract the text. Lastly, the extracted text was fed into the SVM classifier to classify the shop. Figure 6.10 illustrates the whole cycle of the mentioned process on a sample street view image from the SVT dataset.

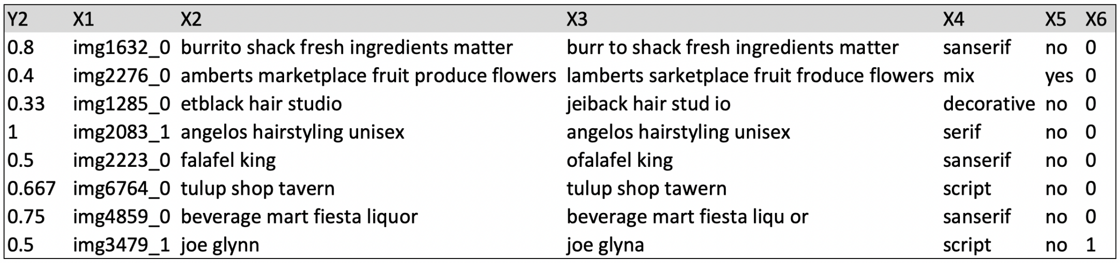
The final accuracy of the tested SVT dataset on our framework reached 84.85% with F1-score equals to 89%. Through all stages, the detector showed robust performance even with difficult angles of street views and occluded signs. Furthermore, the OCR produced good results with low resolution signboard images. When manual verification was performed on the final store classification, it was observed that most of the wrongly classified samples were due to the existence of confusing words. For example, a "restaurant and drink" store that has this text extracted from its signboard "italian espressi shot plus tax" was classified as a "finance and investing" store because of the the word "tax" in addition to the mistake in recognizing the letter "o" in the word "espresso".

Chapter 7

Typeface Influence Analysis

The style of typeface that appears on shop signboards plays an important role in the store classification stage. The shop classification results might be negatively impacted if the signboard's text has not been recognized correctly because of the font design. Since each typeface style included in this study has different design characteristics, we analyzed the influence of these styles on the recognition process. This was done using the outcomes of the experiment explained in Section 5.3. In particular, when the Google Cloud Vision OCR was fed with the ShoS-cropped dataset (i.e. signboard images), it extracted text in an editable form. Then, the extracted text for each signboard was evaluated using two methods LD and IAA as described in Section 6.2 in Chapter 6. For the following typeface influence analysis, only IAA scores were considered and referred by Y2. This is because LD scores take into account the whole text matching while the IAA scores only focus on the essential keywords that would influence the store classification accuracy.

In this chapter, the data used for analyzing the typeface influence will be defined in Section 7.1 and normalized in Section 7.2. Then, the statistical analysis utilized to identify the impact of type styles on the recognition process will be provided in Section 7.3.



Y2	X1	X2	X3	X4	X5	X6
0.8	img1632_0	burrito shack fresh ingredients matter	burr to shack fresh ingredients matter	sanserif	no	0
0.4	img2276_0	amberts marketplace fruit produce flowers	lamberts sarketplace fruit froduce flowers	mix	yes	0
0.33	img1285_0	etblack hair studio	jeiback hair stud io	decorative	no	0
1	img2083_1	angelos hairstyling unisex	angelos hairstyling unisex	serif	no	0
0.5	img2223_0	falafel king	ofalafel king	sanserif	no	0
0.667	img6764_0	tulup shop tavern	tulup shop tawern	script	no	0
0.75	img4859_0	beverage mart fiesta liquor	beverage mart fiesta liqu or	sanserif	no	0
0.5	img3479_1	joe glynn	joe glyna	script	no	1

Figure 7.1: Some samples from the data used to analyze the influence of the type styles on the recognition process

7.1 Data Definition

Defining the data was the first step to proceed with the statistical analysis. Our data is based on the ShoS annotation file merged with the OCR outcomes. Table 7.1 shows the data definition and Figure 7.1 includes some samples from the used data. In total, there are 10k instances where each one presents a shop signboard. We ignored some attributes, such as bounding box coordinates, as they do not add value at this step. The descriptives of Y2: OCR scores across X4: typeface style, X5: occlusion, and X6: difficulty are shown in Table 7.2 to have an indication on the overall distribution of the studied variables. There were no missing values as shown in the attribute "Missing".

Table 7.1: Data definition for the typeface influence analysis

Attribute	**Name**	**Definition**
Score	Y2	The dependent variable (contentious number between 0 and 1) which represents the OCR evaluation score (1=perfect match)
File Name	X1	A unique ID for each image
GT Text	X2	The ground truth text (cleaned)
OCR Text	X3	The extracted text by OCR (cleaned)
Typeface Style	X4	The independent variable (nominal) that has six possible values: Serif, Sanserif, Script, Decorative, Blackletter, Mix
Occluded	X5	The independent variable (dichotomous) that has two possible value: yes, no
Difficulty Level	X6	The independent variable (nominal) that has five possible values, but they are normalized into: 0=not difficult, 1=difficult as explained in data normalization in Section 7.2

Table 7.2: The descriptives of Y2: OCR scores across X4: typeface style, X5: occlusion, and X6: difficulty before data normalization

	X4	X5	X6	N	Missing	Mean	SD	Variance
Y2	blackletter	no	0	15	0	0.721	0.304	0.0927
			1	8	0	0.625	0.443	0.1964
		yes	0	1	0	1	NaN	NaN
			1	1	0	1	NaN	NaN
	decorative	no	0	346	0	0.695	0.367	0.1347
			1	7	0	0.488	0.426	0.1815
		yes	0	25	0	0.576	0.424	0.1802
			1	1	0	0.667	NaN	NaN
	mix	no	0	1806	0	0.797	0.246	0.0604
			1	13	0	0.64	0.334	0.1118
		yes	0	126	0	0.674	0.275	0.0758
			1	3	0	0.85	0.132	0.0175
	sanserif	no	0	4601	0	0.854	0.262	0.0689
			1	62	0	0.811	0.304	0.0924
		yes	0	403	0	0.728	0.32	0.1025
			1	8	0	0.676	0.357	0.1273
	script	no	0	284	0	0.715	0.357	0.1272
			1	5	0	0.6	0.253	0.0639
		yes	0	23	0	0.476	0.454	0.2065
			1	0	0	NaN	NaN	NaN
	serif	no	0	2041	0	0.862	0.263	0.0693
			1	30	0	0.759	0.319	0.1018
		yes	0	188	0	0.775	0.275	0.0758
			1	3	0	0.533	0.503	0.2533

7.2 Data Normalization

Before analyzing the response variable Y2 across all groups of the typeface styles, some data normalization was carried out. First, two type styles were eliminated: “Blackletter” and “Mix”. This is because the “Blackletter” style represents less than 1% of the data which represents an insignificant portion of the total population. In addition, “Mix” style does not have consistency in the number of font styles within its group. So, all samples that are labeled with these two styles were list-wise deleted.

Next, we took into consideration two important factors that might affect the Y2 scores which are occlusion factor X5 and difficulty factor X6. For the dichotomous occlusion factor (X5: yes/no), we ran a t-test and observed a significant association between Y2 and X5 ($t9973 = 11.2$, $p < 0.001$). Tables 7.3 and 7.4 show the t-test result and the group descriptives of Y2 and X5. In addition, Figure 7.2 illustrates the confidence interval for the means at 95%. Hence, all occluded samples (value=yes) were list-wise removed from the data.

Table 7.3: Independent T-Test of Y2: OCR scores and X5: occlusion factor

%	Statistic	df	*p*	Mean difference	SE difference	95% Confidence Interval	
						Lower	Upper
Y2	11.2	9973	<.001	0.116	0.0103	0.0954	0.136

Table 7.4: Group Descriptives of Y2: OCR scores and X5: occlusion factor

	Group	N	Mean	Median	SD	SE
Y2	no	9195	0.833	1.00	0.272	0.00284
	yes	780	0.717	0.778	0.317	0.0113

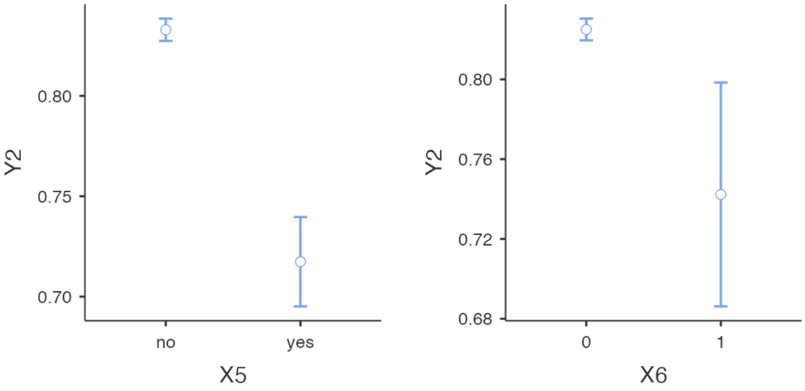


Figure 7.2: The Confidence Interval of means at 95% for X5: occlusion factor (left) and X6: difficulty factor (right)

Furthermore, the difficulty factor X6 was normalized too. It has been elaborated before in Section 4.2 and Table 4.2 that X6 has five scales where each scale represents a different level of difficulty. At this stage, the only concern is if the signboard is difficult or not. Thus, we converted the difficulty factor X6 from a nominal variable into dichotomous with two values only: 1) difficult (value=1), or 2) not difficult (value=0). Then, we ran a t-test and observed a significant association between Y2 and X6 ($t9973 = 3.41$, $p < 0.001$). Tables 7.5 and 7.6 show the t-test result and the group descriptives. Also, Figure 7.2 illustrates the confidence interval for the means at 95%. Therefore, all the difficult samples (value=1) were list-wise eliminated from the data.

Table 7.5: Independent T-Test of Y2: OCR scores and X6: difficulty factor

%	Statistic	df	*p*	Mean difference	SE difference	95% Confidence Interval	
						Lower	Upper
Y2	3.41	9973	<.001	0.0827	0.0243	0.0351	0.130

Table 7.6: The Group descriptives of Y2: OCR scores and X6: difficulty factor

	Group	N	Mean	Median	SD	SE
Y2	0	9843	0.825	1.00	0.276	0.00279
	1	132	0.742	1.00	0.326	0.0284

Table 7.7: The descriptives of Y2: OCR scores across X4: typeface style, X5: occlusion, and X6: difficulty after data normalization

	X4	X5	X6	N	Missing	Mean	SD	Variance
Y2	decorative	no	0	346	0	0.695	0.367	0.1347
	sanserif			4601	0	0.854	0.262	0.0689
	script			284	0	0.715	0.357	0.1272
	serif			2041	0	0.862	0.263	0.0693
				7272	0	0.782	-	-

This way the data were ready for the statistical analysis. The data descriptives after normalization are shown in Table 7.7 where the total number of samples was reduced to 7272. Figure 7.3 visualizes the distribution of Y2 scores for each typeface style in addition to mean points using violin plot.

7.3 Statistical Analysis

We tested the normality using D'Agostino's Kˆ2 statistical test with $alph = 0.05$ and the $p - value$ were much smaller than the alpha. The histogram for the OCR scores Y2 is shown in Figure 7.4(a). It is highly skewed to the left where many of the Y2 values are close to 1, which represents the strength of the used OCR. Figure 7.5 provides a closer look to the values of Y2 for each group in the typeface style X4. We tried to transform data to fix the data distribution using

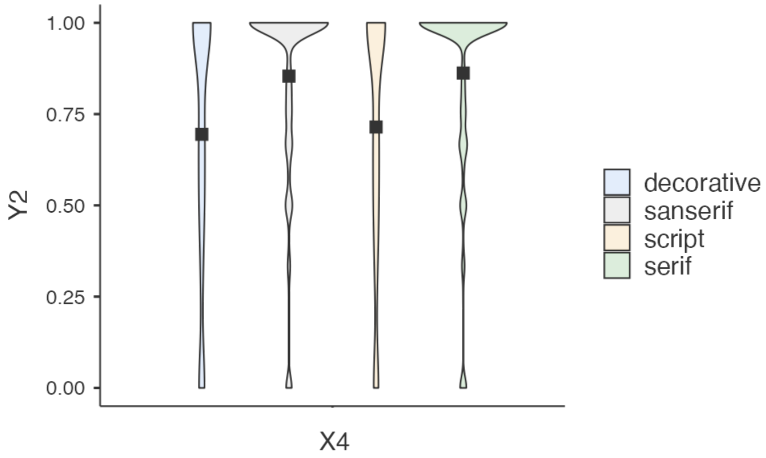
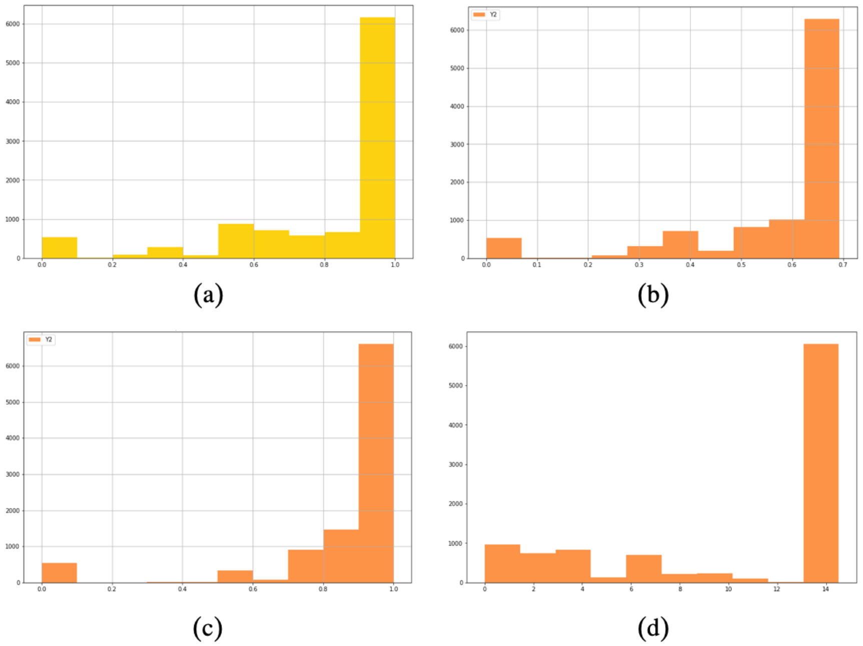


Figure 7.3: The violin plot of Y2 scores for each typeface style in X4 showing data distribution and mean points

three different methods: log, square root, and Cox Box with no success. Figure 7.4 shows the histogram before and after transformation methods. Even in the best scenario case which is the Cox Box method, the data is still skewed. Therefore, it is concluded that the data is not normally distributed. This was because of the high accuracy of the used OCR which led to having more high scores.

Furthermore, we tested data homogeneity, which tests if variances are equal across the groups, using Levene's test [105]. The p-value was highly significant $p < 0.001$, thus we had to reject the null hypothesis and conclude that our data is heterogeneous.

Based on normality and homogeneity tests, ANOVA statistic could not be applied as it requires normality and homogeneity of the data. Instead, a non-parametric test (Kruskal Wallis) was conducted to compare the OCR scores across the typeface styles: Serif, Sanserif, Script, and Decorative. We observed a significant association between Y2 and X4 ($\chi^2(3) = 133$, $p < 0.001$) (see Table 7.8). Figure 7.6 illustrates the confidence interval in means for each type style at 95%. That means the typeface style had impacted the OCR scores which supports our hypothesis that the typeface style used in store signboards affects the recognition performance and hence the store classification



(a)

(b)

(c)

(d)

Figure 7.4: The histogram of the response variable Y2. (a) before transformation, (b) log transformation, (c) square root transformation, and (d) Cox Box transformation

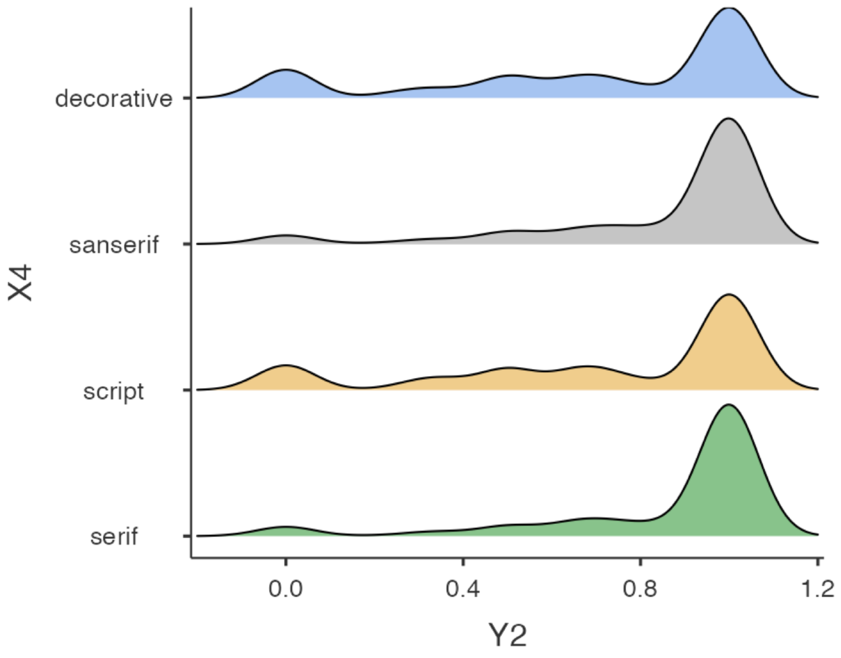


Figure 7.5: The density graph of Y2 for each type style in X4

Table 7.8: Kruskal Wallis analysis for Y2: OCR scores and X4: typeface style

	χ^2	**df**	***p***
Y2	133	3	<.001

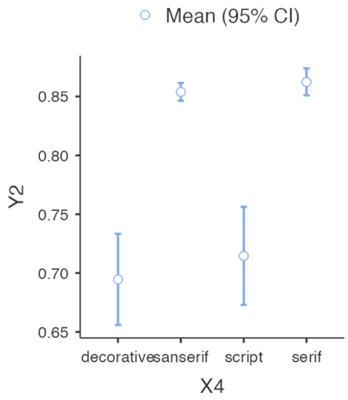


Figure 7.6: The Confidence Interval of means at 95% for X4: typeface style

performance.

Table 7.9: The pairwise comparisons using DSCF post-hoc test for Y2: OCR scores across X4: typeface style

Pair Group		**W**	***p***
decorative	sanserif	12.231	<.001
decorative	script	0.989	0.898
decorative	serif	13.027	<.001
sanserif	script	-9.719	<.001
sanserif	serif	3.306	0.090
script	serif	10.628	<.001

To dig more in the result and find out which group is most effective, a post-hoc test was carried out using Dwass-Steel-Critch-Fliger DSCF pairwise comparisons. Table 7.9 shows the results of DSCF test. The findings indicated significant differences in wise reasoning scores between all groups except Sanserif-Serif and Decorative-Script. Looking at the confidence interval in Figure 7.6, it is clear that Serif followed by Sanserif are falling within high mean range compared to Script and Decorative. Based on these results, Serif and Sanserif font styles are similar to each other in their performance and hence they are the most preferred styles as they showed robust performance against OCR recognition. Therefore, it is recommended to avoid using the type styles Script and Decorative in store signboard because of the sensitivity of the recognition rates to such font styles. If using stylized fonts is a must, it is advised to add key words that distinguish a store class from another using Serif or Sanserif styles.

Chapter 8

Conclusion and Future work

Many factors can influence the process of detecting and classifying commercial retails based on their visual appearance. Previous studies built models that considered the whole storefronts however, that has negatively affected the classification process. This was because the inclusion of other components within the storefronts. This research focuses on shop signboards as they are much more consistent than the whole storefront. In this chapter, a summary of this thesis' significant contributions will be provided in Section 8.1. In addition, topics for future work will be listed in Section 8.2. In particular, some areas can be enhanced for further improvement in the provided work to support real-world street systems. Finally, the researcher's publications regarding this work will be listed in section 8.3.

8.1 Summary of Contributions

In this thesis, we introduced the ShoS dataset, the Shop Signboard Detection and Classification Framework SSDCF to detect and classify commercial stores, and the analysis of typeface design used in shop signboards. The framework was fully implemented and its performance was evaluated using different measurements. Figure 8.1 represents the flow of activities of this thesis starting from dataset collection, store signboard detection, text extraction, store classification, SSDCF evaluation, and typeface impact analysis.

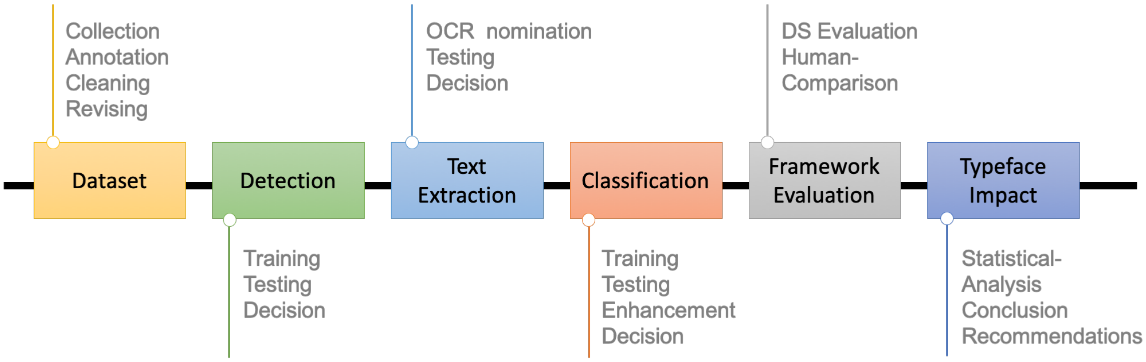


Figure 8.1: Thesis flow of activities

The ShoS dataset was collected from Google street images. A total of 10k store signboards were captured within 7500 storefront images. All the collected images were fully annotated and made for the public. The annotation of the ShoS dataset includes different attributes for each image and for each signboard, such as bounding box coordinates, store class, and typeface style. The ShoS and the ShoS-cropped datasets can be used for several research purposes including store signboard detection, store classification, text recognition, and typeface classification.

A framework was designed and built with three main components: 1) the detector to detect signboards of shops, 2) the text extractor to extract text from the detected signboards, and 3) the classifier to classify commercial stores based on the textual information.

For signboard detection, two models were trained and tested utilizing the ShoS dataset. Findings surpassed the performance of YOLOv3 for signboard detection. It was noticed that the detector performed better when it was trained on true negative samples. Also, converting the color scheme into Grayscale did not result in higher mean average precision. Thus, it is not necessary to pre-process the color scheme of the input image.

For text extraction, the evaluation of Google Vision OCR showed better results even with the existence of influential factors, such as stylized fonts and skewed images. That might be attributed to the huge amount of data that Google has for building and training its OCR.

For store classification, two models were trained and tested utilizing the ShoS dataset. SVM showed great performance even with classes that have a lower number of samples and a high number of non-descriptive text like classes “Fashion” and “Technology”. The performance of the classifier had been enhanced by 4% approximately after adding the augmented data. The augmented data was generated by utilizing the Random Deletion method and Thesauruses-inspired method named *OCR-Thesauruses*.

The full framework was evaluated using the SVT dataset [10], and the outcome reached an accuracy of 84.85% with F1-score equals to 89%. The classification performance was also compared with human performance, and the results showed that our classifier excelled over human performance by about 15%. Most of the misclassified samples were coupled with the ambiguity factor, so this highlights the importance of adding descriptive keywords related to store class in the signboards in order to increase the accuracy of classifying stores by humans and machines.

Finally, the results coming from the second component of our framework, the text extractor, were statistically analyzed to check the impact of typeface styles used in shop signboards on the recognition rates. The findings showed a significant association between the typeface style and the recognition rate. Based on the analysis, it is recommended to use “Serif” and “Sanserif” styles over “Script” and “Decorative” in shop signboards as they provide a higher performance. If using stylized fonts is a must for showing a unique identity, it is advised to add key words that distinguish a store class from another using “Serif” or “Sanserif” styles.

8.2 Future Work

This work can be enhanced in the following directions. The current version of the ShoS dataset is limited to English shop signboards. It can be enhanced by including more common languages like French, Chinese, and Arabic. In addition, the scope of store super-classes could be extended to include more than six super classes for store classification, such as gas stations and educational institutes. This would require collecting and annotating more data where our trained detector can be utilized to annotate the bounding boxes automatically. Expanding the scope of language and super classes would assure the generalization of our framework.

Furthermore, the detector can be trained and tested on some street views that were taken at night or with dark lighting to mimic real scenarios. Also, the classifier may use other features, such as color or shape to get more clues about the store type by applying ensemble techniques where multiple models are combined to find the right class.

For the typeface part, some standard recommendations would be introduced based on a deep analysis of the visual appearance of signboards to formalize them for better legibility and recognition. In particular, special fonts could be designed and produced for business owners to ensure readability by humans and machines while keeping harmony with the surrounding area.

8.3 Publications

Through the journey of my PhD, the following publications have been published:

- Mrouj Almuhajri, and Ching Suen. "Intensive Survey About Road Traffic Signs Preprocessing, Detection and Recognition." International Conference on Computing. Springer, Riyadh, Saudi Arabia 2019. [25]

- Mrouj Almuhajri and Ching Y. Suen. 2022. "Shop Signboards Detection Using the ShoS Dataset." In Pattern Recognition and Artificial Intelligence: Third International Conference, ICPRAI 2022, Paris, France, June 1–3, 2022, Proceedings, Part II. Springer-Verlag, Berlin, Heidelberg, pp 235–245. [106]

- Mrouj Almuhajri and Ching Y. Suen. 2022. “A Complete Framework for Shop Signboards Detection and Classification”. In International Conference on Pattern Recognition ICPR 2022, Montreal, Canada, August 21-25, 2022. (in-press).
- Mrouj Almuhajri and Ching Y. Suen. 2022. “AI Based Approach for Shop Classification with a Comparative Study with Human”. In *Advances in Artificial Intelligence and Machine Learning AAIML Journal*, (accepted).

Bibliography

[1] Building and planning department of the City of Westmount. Renovating and building in westmount - storefronts and signage. https://www.westmount.org/wp-content/uploads/2014/07/7-Storefronts_and_Signage.pdf, Sep 2001. Online; accessed 30 March 2020.

[2] Fei-Fei Li, Andrej Karpathy, and Justin Johnson. Lecture 8: Spatial localization and detection. http://cs231n.stanford.edu/slides/2016/winter1516_lecture8.pdf, February 2016. Online; accessed 5 May 2020.

[3] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):142–158, 2016.

[4] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, Santiago, Chile, 2015.

[5] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015.

[6] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, abs/1804.02767, 2018.

[7] Hyacinth Ampadu. Yolov3 and yolov4 in object detection. https://ai-pool.com/a/s/yolov3-and-yolov4-in-object-detection, 2021. Online; accessed 10 May 2021.

[8] Lilian Weng. Object detection part 4: Fast detection models. https://lilianweng.github.io/lil-log/2018/12/27/object-detection-part-4.html#yolov3, 2018. Online; accessed 28 April 2020.

[9] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37, Amsterdam, Netherlands, 2016. Springer.

[10] Kai Wang and Serge Belongie. Word spotting in the wild. In *European Conference on Computer Vision*, pages 591–604, Crete, Greece, 2010. Springer.

[11] Qian Yu, Christian Szegedy, Martin C Stumpe, Liron Yatziv, Vinay Shet, Julian Ibarz, and Sacha Arnoud. Large scale business discovery from street level imagery. *arXiv preprint arXiv:1512.05430*, 2015.

[12] Shahin Sharifi Noorian, Sihang Qiu, Achilleas Psyllidis, Alessandro Bozzon, and Geert-Jan Houben. Detecting, classifying, and mapping retail storefronts using street-level imagery. In *International Conference on Multimedia Retrieval (ICMR)*, pages 495–501, Dublin, Ireland, 06 2020.

[13] Yair Movshovitz-Attias, Qian Yu, Martin C Stumpe, Vinay Shet, Sacha Arnoud, and Liron Yatziv. Ontological supervision for fine grained classification of street view storefronts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1693–1702, Boston, MA, USA, 2015.

[14] Stefano Messelodi, Carla Maria Modena, Lorenzo Porzi, and Paul Chippendale. i-street: Detection, identification, augmentation of street plates in a touristic mobile application. In *International Conference on Image Analysis and Processing*, pages 194–204. Springer, 2015.

[15] Vivian Cook. *The Language of the English Street Sign*. Multilingual Matters, 2022.

[16] Yi Yang, Hengliang Luo, Huarong Xu, and Fuchao Wu. Towards real-time traffic sign detection and classification. *IEEE Transactions on Intelligent Transportation Systems*, 17(7):2022–2031, 2016.

[17] Po-Cheng Shih, Chi-Yi Tsai, and Chun-Fei Hsu. An efficient automatic traffic sign detection and recognition method for smartphones. In *Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), 2017 10th International Congress on*, pages 1–5. IEEE, 2017.

[18] Jack Greenhalgh and Majid Mirmehdi. Traffic sign recognition using mser and random forests. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pages 1935–1939. IEEE, 2012.

[19] HG Parsa, Jean-Pierre I van der Rest, Scott R Smith, Rahul A Parsa, and Milos Bujisic. Why restaurants fail? part iv: The relationship between restaurant failures and demographic factors. *Cornell Hospitality Quarterly*, 56(1):80–90, 2015.

[20] Alexander W Bartik, Marianne Bertrand, Zoe Cullen, Edward L Glaeser, Michael Luca, and Christopher Stanton. The impact of covid-19 on small business outcomes and expectations. *Proceedings of the National Academy of Sciences*, 117(30):17656–17666, 2020.

[21] Paco Underhill. How to read a sign. In *Why we buy: The science of shopping–updated and revised for the Internet, the global consumer, and beyond*, chapter 5, pages 74–89. Simon and Schuster, 2009.

[22] Bryan Reimer, Bruce Mehler, Jonathan Dobres, Joseph F Coughlin, Steve Matteson, David Gould, Nadine Chahine, and Vladimir Levantovsky. Assessing the impact of typeface design in a text-rich automotive user interface. *Ergonomics*, 57(11):1643–1658, 2014.

[23] Rabiah Al-qudah and Ching Y. Suen. Impact of font on computer recognition of license plates on automobiles. In *Proceedings of the 3rd International Conference on Vision, Image and Signal Processing*, ICVISP 2019, New York, NY, USA, 2019. Association for Computing Machinery.

[24] Lukáš Neumann and Jiří Matas. Real-time scene text localization and recognition. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3538–3545. IEEE, 2012.

[25] Mrouj Almuhajri and Ching Suen. Intensive survey about road traffic signs preprocessing, detection and recognition. In *International Conference on Computing*, pages 275–289, Riyadh, Saudi Arabia, 2019. Springer.

[26] Dong Yan, Wang Li, and Wang YingZhi. The design of commercial signboard fonts in shenyang to establish urban visual orders. In *E3S Web of Conferences*, volume 179, page 02065. E3S Web of Conferences, 2020.

[27] Charles Bigelow. Typeface features and legibility research. *Vision research*, 165:162–172, 2019.

[28] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, Columbus, Ohio, USA, 2014.

[29] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916, 2015.

[30] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *ArXiv*, abs/1905.05055, 2019.

[31] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.

[32] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015.

[33] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.

[34] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection, 2020.

[35] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[36] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[37] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, Venice, Italy, 2017.

[38] Mark Everingham and John Winn. The pascal visual object classes challenge 2007 (voc2007) development kit. *University of Leeds, Tech. Rep*, 2007.

[39] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755, Zürich, Switzerland, 2014. Springer.

[40] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7310–7311, Honolulu, HI, USA, 2017.

[41] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2147–2154, Columbus, Ohio, USA, 2014.

[42] Xiaoxue Chen, Lianwen Jin, Yuanzhi Zhu, Canjie Luo, and Tianwei Wang. Text recognition in the wild: A survey. *ACM Computing Surveys (CSUR)*, 54(2):1–35, 2021.

[43] Chhanam Thorat, Aishwarya Bhat, Padmaja Sawant, Isha Bartakke, and Swati Shirsath. A detailed review on text extraction using optical character recognition. In Simon Fong, Nilanjan Dey, and Amit Joshi, editors, *ICT Analysis and Applications*, pages 719–728, Singapore, 2022. Springer Singapore.

[44] ABBYY FineReader. https://www.abbyy.com/ocr-sdk/ocr-stages/, 2021. Online; accessed 01 March 2021.

[45] Google Cloud. Cloud vision api. https://cloud.google.com/vision/docs/ocr, 2021. Online; accessed 01 March 2021.

[46] Vladimir I Levenshtein et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union, 1966.

[47] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. Text classification algorithms: A survey. *Information*, 10(4):150, 2019.

[48] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing Management*, 24(5):513–523, 1988.

[49] M I Jordan. Serial order: a parallel distributed processing approach. technical report, june 1985-march 1986. 5 1986.

[50] V Vapnik and A Ya Chervonenkis. A class of algorithms for pattern recognition learning. *Avtomat. i Telemekh*, 25(6):937–945, 1964.

[51] Eibe Frank and Remco R. Bouckaert. Naive bayes for text classification with unbalanced classes. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Knowledge Discovery in Databases: PKDD 2006*, pages 503–510, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[52] Shigeo Abe. Analysis of multiclass support vector machines. *Thyroid*, 21(3):3772.

[53] Vasileios Milias and Achilleas Psyllidis. Assessing the influence of point-of-interest features on the classification of place categories. *Computers, Environment and Urban Systems*, 86:101597, 2021.

[54] Ilene Strizver. What makes a typeface look the way it does? In *Type Rules: The designer's guide to professional typography*, chapter 3, pages 37–49. John Wiley & Sons, 2013.

[55] Chongsheng Zhang, Guowen Peng, Yuefeng Tao, Feifei Fu, Wei Jiang, George Almpanidis, and Ke Chen. Shopsign: a diverse scene text dataset of chinese shop signs in street views. *arXiv preprint arXiv*, abs/1903.10412, 2019.

[56] Md. Sadrul Islam Toaha, Chowdhury Rafeed Rahman, Sakib Bin Asad, Tashin Ahmed, M. A. Proma, and S. Haque. Automatic signboard detection from natural scene image in context of bangladesh google street view. *ArXiv*, abs/2003.01936, 2020.

[57] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. Icdar 2013 robust reading competition. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1484–1493, Washington, DC, USA, 2013. IEEE.

[58] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1156–1160, Nancy, France, 2015. IEEE.

[59] Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*, 2016.

[60] Tai-Ling Yuan, Zhe Zhu, Kun Xu, Cheng-Jun Li, Tai-Jiang Mu, and Shi-Min Hu. A large chinese text dataset in the wild. *Journal of Computer Science and Technology*, 34(3):509–521, 2019.

[61] Ying Zhang, Lionel Gueguen, Ilya Zharkov, Peter Zhang, Keith Seifert, and Ben Kadlec. Uber-text: A large-scale dataset for optical character recognition from street-level imagery. In *SUNw: Scene Understanding Workshop - CVPR 2017*, Hawaii, U.S.A., 2017.

[62] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[63] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015.

[64] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, Boston, MA, USA, 2015.

[65] Stefan Funke and Sabine Storandt. Automatic tag enrichment for points-of-interest in open street map. In David Brosset, Christophe Claramunt, Xiang Li, and Tianzhen Wang, editors, *Web and Wireless Geographical Information Systems*, pages 3–18, Cham, 2017. Springer International Publishing.

[66] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Gonçalves, W. R. Schwartz, and D. Menotti. A robust real-time automatic license plate recognition based on the yolo detector. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10, Rio, Brazil, 2018.

[67] Rabiah Al-qudah and Ching Y Suen. Enhancing yolo deep networks for the detection of license plates in complex scenes. In *Proceedings of the Second International Conference on Data Science, E-Learning and Information Systems*, pages 1–6, Dubai, UAE, 2019.

[68] J. Ren and H. Li. Implementation of vehicle and license plate detection on embedded platform. In *2020 12th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, pages 75–79, Phuket, Thailand, 2020.

[69] Hoanh Nguyen. Real-time license plate detection based on vehicle region and text detection. *Journal of Theoretical and Applied Information Technology*, 98(03), 2020.

[70] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and H Adam Mobilenets. Efficient convolutional neural networks for mobile vision applications. *arXiv preprint ArXiv:1704.0486*, 2017.

[71] Yingying Zhu, Cong Yao, and Xiang Bai. Scene text detection and recognition: Recent advances and future trends. *Frontiers of Computer Science*, 10(1):19–36, 2016.

[72] Shangbang Long, Xin He, and Cong Yao. Scene text detection and recognition: The deep learning era. *International Journal of Computer Vision*, 129(1):161–184, 2021.

[73] Yipeng Sun, Chengquan Zhang, Zuming Huang, Jiaming Liu, Junyu Han, and Errui Ding. Textnet: Irregular text reading from images with an end-to-end trainable network. In C. V. Jawahar, Hongdong Li, Greg Mori, and Konrad Schindler, editors, *Computer Vision – ACCV 2018*, pages 83–99, Cham, 2019. Springer International Publishing.

[74] Alessandro Bissacco, Mark Cummins, Yuval Netzer, and Hartmut Neven. Photoocr: Reading text in uncontrolled conditions. In *Proceedings of the IEEE international conference on computer vision*, pages 785–792, 2013.

[75] Ahmad P Tafti, Ahmadreza Baghaie, Mehdi Assefi, Hamid R Arabnia, Zeyun Yu, and Peggy Peissig. Ocr as a service: an experimental evaluation of google docs ocr, tesseract, abbyy finereader, and transym. In *International Symposium on Visual Computing*, pages 735–746. Springer, 2016.

[76] Melvin Lundqvist and Agnes Forsberg. A comparison of ocr methods on natural images in different image domains. KTH Royal Instiute of Technology, School of Electrical Engineering and Computer Science, Stockholm, Sweden, 2020.

[77] Robert Nagy, Anders Dicker, and Klaus Meyer-Wegener. Neocr: A configurable dataset for natural image text recognition. In *International Workshop on Camera-Based Document Analysis and Recognition CBDAR*, pages 53–58, Beijing, China, 2011. Springer.

[78] Robert Nagy, Anders Dicker, and Klaus Meyer-Wegener. Definition and evaluation of the neocr dataset for natural-image text recognition. Technical report, University of Erlangen, Dept. of Computer Science, CS-2011-07, 2011.

[79] Yulia S Chernyshova, Alexander V Sheshkus, and Vladimir V Arlazarov. Two-step cnn framework for text line recognition in camera-captured images. *IEEE Access*, 8:32587–32600, 2020.

[80] VV Arlazarov, K Bulatov, T Chernov, and VL Arlazarov. A dataset for identity documents analysis and recognition on mobile devices in video stream. *Comput. Opt.*, 43:818–824, 2019.

[81] Rafsanjany Kushol, Imamul Ahsan, and Md Nishat Raihan. An android-based useful text extraction framework using image and natural language processing. *International Journal of Computer Theory and Engineering*, 10(3):77–83, 2018.

[82] Akshat Pathak, Aviral Ruhela, Anshul K Saroha, and Anant Bhardwaj. Examining robustness of google vision api based on the performance on noisy images. *International Journal of Computer Sciences and Engineering JCSE*, 7(3):89–93, 2019.

[83] Weslley Torres, Mark GJ van den Brand, and Alexander Serebrenik. Suitability of optical character recognition (ocr) for multi-domain model management. In *International Conference on Systems Modelling and Management*, pages 149–162, Bergen, Norway, 2020. Springer.

[84] Ammar Ismael Kadhim. Survey on supervised machine learning techniques for automatic text classification. *Artificial Intelligence Review*, 52(1):273–292, 2019.

[85] Fanny Fanny, Yohan Muliono, and Fidelson Tanzil. A comparison of text classification methods k-nn, naïve bayes, and support vector machine for news classification. *Jurnal Informatika: Jurnal Pengembangan IT*, 3(2):157–160, 2018.

[86] Md Ataur Rahman and Yeasmin Ara Akter. Topic classification from text using decision tree, k-nn and multinomial naïve bayes. In *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, pages 1–4. IEEE, 2019.

[87] Niken Larasati Octaviani, Eko Hari Rachmawanto, Christy Atika Sari, and Ignatius Moses Setiadi De Rosal. Comparison of multinomial naïve bayes classifier, support vector machine, and recurrent neural network to classify email spams. In *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*, pages 17–21. IEEE, 2020.

[88] Jehan Janbi, Mrouj Almuhajri, and Ching Y Suen. The effect of fonts design features on ocr for latin and arabic. *Journal ISSN*, 2368:5956, 2014.

[89] Mrouj Almuhajri and Ching Y Suen. Legibility and readability of arabic fonts on personal digital assistants pdas. In *Digital Fonts and Reading*, pages 248–265. World Scientific, 2016.

[90] Nafiseh Hojjati and Balakrishnan Muniandy. The effects of font type and spacing of text for online readability and performance. *Contemporary Educational Technology*, 5(2):161–174, 2014.

[91] Sofie Beier and Chiron A.T. Oderkerk. High letter stroke contrast impairs letter recognition of bold fonts. *Applied Ergonomics*, 97:103499, 2021.

[92] Roland Schroll, Benedikt Schnurr, and Dhruv Grewal. Humanizing products with handwritten typefaces. *Journal of Consumer Research*, 45(3):648–672, 2018.

[93] Stephanie Q Liu, Sungwoo Choi, and Anna S Mattila. Love is in the menu: Leveraging healthy restaurant brands with handwritten typeface. *Journal of Business Research*, 98:289–298, 2019.

[94] Abhishek Dutta and Andrew Zisserman. The VIA annotation software for images, audio and video. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM ’19, New York, NY, USA, 2019. ACM.

[95] A. Dutta, A. Gupta, and A. Zissermann. VGG image annotator (VIA). http://www.robots.ox.ac.uk/ vgg/software/via/, 2016. Version: X.Y.Z, Accessed: 15 Sep 2018.

[96] Dragomir Anguelov, Carole Dulong, Daniel Filip, Christian Frueh, Stéphane Lafon, Richard Lyon, Abhijit Ogale, Luc Vincent, and Josh Weaver. Google street view: Capturing the world at street level. *Computer*, 43(6):32–38, 2010.

[97] Xuemei Xie, Xun Xu, Lihua Ma, Guangming Shi, and Pengfei Chen. On the study of predictors in single shot multibox detector. In *Proceedings of the International Conference on Video and Image Processing*, ICVIP 2017, pages 186–191, New York, NY, USA, 2017. ACM.

[98] M. Namysl and I. Konya. Efficient, lexicon-free ocr using deep learning. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 295–301, 2019.

[99] Yulia S Chernyshova, Alexander V Sheshkus, and Vladimir V Arlazarov. Two-step cnn framework for text line recognition in camera-captured images. *IEEE Access*, 8:32587–32600, 2020.

[100] Bohan Li, Yutai Hou, and Wanxiang Che. Data augmentation approaches in natural language processing: A survey. *AI Open*, 2022.

[101] Sridevi Bonthu, Abhinav Dayal, M. Sri Lakshmi, and S. Rama Sree. Effective text augmentation strategy for nlp models. In Ramesh Chandra Poonia, Vijander Singh, Dharm Singh Jat, Mario José Diván, and Mohammed S. Khan, editors, *Proceedings of Third International Conference on Sustainable Computing*, pages 521–531, Singapore, 2022. Springer Nature Singapore.

[102] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*, 2019.

[103] Adam Kilgarriff. Thesauruses for natural language processing. In *International Conference on Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003*, pages 5–13. IEEE, 2003.

[104] Cheyna Katherine Brower. Too long and too boring: The effects of survey length and interest on careless responding, 2018. Wright State University.

[105] Ingram Olkin, Sudhish G Ghurye, Wassily Hoeffding, William G. Madow, and Henry B. Mann. *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*. Stanford University Press, pp. 278-292, 1960.

[106] Mrouj Almuhajri and Ching Y. Suen. Shop signboards detection using the shos dataset. In Mounîm El Yacoubi, Eric Granger, Pong Chi Yuen, Umapada Pal, and Nicole Vincent, editors, *Pattern Recognition and Artificial Intelligence*, pages 235–245, Cham, 2022. Springer International Publishing.