Recognition and Prominence Ranking of Alphanumeric Number Sequences in Images

Alex Cummaudo
BSc *Swinburne*

Supervised by Prof. Rajesh Vasa, Assoc. Prof. Andrew Cain

A thesis submitted in partial fulfilment of the requirements for the
Bachelor of Information Technology (Honours)

Deakin Software and Technology Innovation Laboratory
School of Information Technology
Deakin University, Australia

October 2017

Abstract

Text detection in natural images is a growing area with increasing applications, including traffic sign and license plate recognition, and text-based image search. Robustly detecting and recognising text is especially challenging when text is deformed, such as the photometric and geometric distortions of text worn by a moving subject in unstructured scenes. Existing methods of text detection in such cases are classified as learning-based or connected component (CC)-based, applying a mix of enhanced detection techniques—such as stroke width transformation (SWT), canny-edge detection and maximally stable extremal regions (MSERs)—and feeding candidates into optical character recognition (OCR) engines or neural networks to recognise the text. This study proposes applying a learning-based approach using deep-learning strategies to automate the recognition of racing bib numbers (RBNs) in a natural image dataset of various marathons, and then ranking detected subject's photos in order of prominence. Experimental results showed that these deep-learning strategies performed favourably against other methods using a consistent dataset, prompting further investigation in the generality of the technique developed to other similar subject material.

Declarations

I certify that the the thesis entitled "Recognition and Prominence Ranking of Alphanumeric Number Sequences in Images" submitted for the degree of Bachelor of Information Technology (Honours) is the result of my own work and that where reference is made to the work of others, due acknowledgement is given. I also certify that any material in the thesis which has been accepted for a degree or diploma by any university of institution is identified in the text.

Alex Cummaudo, BSc *Swinburne*
October 2017

We certify that the thesis prepared by Alex Cummaudo entitled "Recognition and Prominence Ranking of Alphanumeric Number Sequences in Images" is prepared according to our expectations and that the honours coordinator can proceed to accept this submission for examination.

Prof. Rajesh Vasa
October 2017

Assoc. Prof. Andrew Cain
October 2017

Acknowledgements

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Contents

List of Abbreviations

CC Connected Component. 2, 4, 5, 7–12, 16, 19, 22

CNN Convolutional Neural Network. 5, 9, 14, 15

DSTIL Deakin Software and Technology Innovation Laboratory. 4

HOG Histogram of Oriented Gradient. 13, 18, 22

ICDAR International Conference on Document Analysis and Recognition. 16, 20

LBP Local Binary Patterns. 13, 22

LPR License Plate Recognition. 2, 4

MLP Multilayer Perceptron. 13, 17

MSER Maximally Stable Extremal Region. 9, 10, 17, 22

NN Neural Network. 2, 4–7, 13, 17–19, 22, 23

OCR Optical Character Recognition. 1, 2, 5, 7, 16–19, 22

PNN Probabilistic Neural Network. 18

RBN Racing Bib Number. 2–7, 16, 18, 19, 23

SIFT Scale Invariant Feature Transform. 13

SURF Speeded-Up Robust Features. 13

SVM Support Vector Machine. 13, 14, 22

SWT Stroke Width Transformation. 8, 9, 19, 22

TSR Traffic Sign Recognition. 2, 17, 18, 23

List of Figures

List of Tables

Chapter 1

Introduction

Ever since the camera and phone were unified into smartphones, we have seen an increasing interest for image understanding (specifically to identify the content of an image) but text recognition still faces challenges within images of unstructured scenes. While successes in character recognition have a long history with Optical Character Recognition (OCR) engines [101], these are typically applied under strict conditions (e.g., flatbed scanners for documents without distracting backgrounds). Once applied within the context of a natural scene, real-world discrepancies pose serious shortcomings, such as illumination conditions, viewpoint and perspective differences, blur and glare variations, geometric and photometric distortion, and differences in font size and style [51, 119]. Overcoming these issues has motivated a variety of techniques to realise potential applications that make use of text recognition at scale.

With the ubiquity of smartphone cameras, practical applications of natural image processing have increased. In the last two decades, we have seen the development of point-and-shoot product recognition [33, 110], object detection in videos [98], building recognition [108], image feature extraction to improve visual-based search engines [4, 76], and translation services of American Sign Language gestures [47]. Nonetheless, embedded text within images contains indexable data on the image's semantics [99]; if text extraction is therefore not robust, information extraction suffers.

Text detection robustness is a factor which severely limits a text recognition pipeline. Research in overcoming such limitations have been competed numerous competitions [44, 77, 78, 95], where robustness is the key focus in the image processing pipelines proposed. This focus was reiterated by Chen et al. [16], who state the primary prerequisite for text-based recognition (especially within natural scenes) is the text location must be robustly located.

As with any data processing pipeline, false negatives increase where early stages of the pipeline fail, and therefore detection of these potential candidates must be robust. We can reduce errors, and thus robustness, in a pipeline where: (1) there are unwarranted stages (*excluding* unnecessary stages may also assist in reducing error cases) and (2) by piping through unmatched candidates to further pipelines, which can increase the detection.

Without the construct of robustness, we restrict these pipelines to very confined conditions, and its usefulness in products is not warranted. Therefore, the robustness of text extraction pipelines are imperative to gapping the semantic extraction of information from an image [99], and solving this issue can assist in applications of image processing and data indexing of content within images [27] of paramount proportions.

1.1 Background

This study focuses on character recognition in unstructured scenes (Figure 1.1): specifically, short, alphanumeric number sequences. Previous works present methods to extract these sequences in various areas, namely: License Plate Recognition (LPR) systems [2, 11]; Traffic Sign Recognition (TSR) [25, 58, 67, 93]; and, street number recognition, specifically a study by Netzer et al. [85], using Google Street View[1] to determine the numerical value of street numbers. Figure 1.2 highlights typical usage of these sequences.

Different applications apply varying methods to parse short alphanumeric characters. There are typically two stages of any parsing method: *detection* and *recognition*. Detection refers to locating possible candidates and recognition refers to the representation of the text itself. Detection techniques usually are categorised as either Connected Component (CC)-based or learning or texture-based. CC-based detection will typically use a set of distinct properties on the image to detect relevant areas (such as width, stroke and colour) while learning-based feed images into a classifier that can distinguish candidates from false positives. The recognition phase can typically be achieved using Optical Character Recognition (OCR) engines (such as Tesseract[2]) [6], machine learning algorithms [58, 62, 85] or deep-learning Neural Networks (NNs) to classify detected regions [48, 67, 94].

This study proposes the development of a learning-based detection and recognition pipeline using deep-learning neural networks within the context of unstructured photos, with a focus on marathon Racing Bib Numbers (RBNs)[3], as shown in Figure 1.1.

[1] `https://www.google.com/streetview/` last accessed 13 May 2017.
[2] `https://github.com/tesseract-ocr/tesseract` last accessed 14 May 2017.
[3] While referred to as numbers, some RBNs have alphabetic identifiers in them.

Figure 1.1: Four RBNs in a sample marathon photo.

(a) Successful LPR character segmentation [2]. *Left to right*: original image; region segmentation; character segmentation after negation, height and orientation measurements.

(b) Successful recognition of speed sign digits shown in Eichner and Breckon [25].

(c) Localisation of digits found from varying street view house numbers using the worker described in Netzer et al. [85].

Figure 1.2: Various sample alphanumeric sequences observed in literature.

1.2 Motivation

Detection is harder when the photo is unstructured. Early investigations in License Plate Recognition (LPR) systems were systematic in the subject material assessed; a detailed survey by [3] showed that they work best with consistent lighting, specific colour and typeface detection, fixed detection regions, and non-noisy backgrounds. When applied in the context of images with unstructured backgrounds, these systematic approaches begin to have limitations as the text components cannot be easily determined.

While further investigations in the area utilise enhanced Connected Component (CC)-based detection [16, 26, 97], performance is likely to degrade as image complexity increases [65]. This is especially relevant when text is geometrically obfuscated, such as malformed Racing Bib Numbers (RBNs) as worn on a marathon runner's torso. Malformed, in this sense, is caused by non-flat bib sheets that tend to follow the runner's body shape, in addition to images that are taken in dynamical contexts. Some studies have shown to overcome this by using facial recognition to find a more distinct candidate area [6], but nonetheless rely on a person's face to detect a number. Similarly, typical recognition techniques interpret text as segmented characters, rather than a single string, though there are exceptions such as in Zhu et al. [122].

We also identify subject prominence ranking within natural scenes as an area that has little exploration within literature. (For example, the prominence of a *specific* marathon runner within a scene of many runners.) Prominence ranking is an important field in the context of RBN recognition: runners typically choose not to purchase photos where they have been recognised in an image but are not in the foreground. There are also varying factors which influence purchase likelihood, such as face visibility, eye contact with the camera, and blurriness. An assessment into how the prominence of a runner can be ordered in hundreds of identified photos (based from their recognised RBN) can be used by use of a Neural Network (NN).

This study forms part of an industry project under the Deakin Software and Technology Innovation Laboratory (DSTIL). As a part of the research project, access has been made to a labelled dataset of hundreds of thousands of marathon photos.

1.3 Research Goals

This study aims to develop a processing pipeline that both detects and recognises RBNs on a marathon runner, and then ranks the prominence of each runner detected in the photo. The intention is to explore the viability of artificial deep-learning NNs—such as Convolutional Neural Networks (CNNs)—in the pipeline. Previous studies in RBN recognition [6] and similar areas [25, 58, 109] were heavily heuristic and rule driven.

This primary aim is developed into three key objectives:

Goal 1: ***Detect RBNs using a CNN***

Literature has shown that heuristic-based detection algorithms (that are CC-based) are able to detect text within photos [16, 25, 65]. We propose to apply these rule-based techniques to a large labelled dataset within the context of RBNs, and contrast them against a learning-based detection and recognition algorithms (using NNs). By benchmarking a against existing libraries and open source tools, we explore if heuristic-based detection algorithms (focusing namely on CC-based detection) outperforms learning-based detection methods. For this goal the research question is framed as:

RQ1) Do CNNs detect RBNs with equal or higher recall and precision rates than CC-based methods?

Goal 2: ***Design a CNN that can recognise RBNs***

Typically, traditional alphanumeric sequence parsing can be performed by character segmentation, and then piping those characters into Optical Character Recognition (OCR) engines. In the context of marathon photos, we explore answers to the following:

RQ2) Does a CNN-based OCR approach outperform or is at parity with traditional OCR approaches with higher or equal recall and precision rates?

RQ3) Does a CNN-based OCR algorithm perform *without* the use of character segmentation?

Goal 3: ***Rank prominence of alphanumeric sequences***

Our research objective is aimed to compare if humans are always better at ranking the prominence of an RBN than a NN. We can therefore propose the followings research questions:

RQ4) Can a deep-learning NN be trained to rank marathon runners by prominence?, and if so

RQ5) Does a trained deep-learning NN rank prominence of a runner better or equal to a human?

1.4 Thesis Organisation

This thesis is organised into the chapters as outlined below. An appendix follows with additional supplementary material.

Chapter 2 - Background Provides an overview of prior studies broadly around the areas of number detection and recognition in image processing and artificial NNs.

Chapter 3 - Data Set Describes the data set to be used, data treatment steps, possible techniques in closer depth to develop a number recognition pipeline, and explores ways to develop prominence ranking techniques.

Chapter 4 - Benchmarking Collates results of a series of experiments using our dataset amongst existing open source tools and pipelines presented in previous work

Chapter 5 - Processing Pipeline Discusses the proposed processing pipeline developed that satisfies the aims of this study.

Chapter 6 - Findings Outlines the method used for validation and presentation of our results.

Chapter 7 - Discussion Presents implications that were found from the results of our findings and limitations.

Chapter 8 - Conclusions and Future Work Draws a number of conclusions and alleviates gaps in the findings of this work by presenting future studies.

Summary

In this chapter we identified some shortcomings in text recognition, developed the context of the study—namely RBN detection. We discussed the general stages that exist for text parsing within natural scenes, detection and recognition, and introduced typical techniques that are applied in this context. We outlined the research aims this study achieves, and how the thesis is organised. The following chapter will detail applications of image processing, using neural networks for image processing, and outline what techniques have been used in previous studies to achieve this.

Chapter 2

Background

We have introduced the context of image processing and neural networks in the field, and discussed how text capturing within photos is typically achieved in two stages: detection and recognition. Detection techniques are classified as either CC- or learning-based. The recognition phase uses traditional OCR engines or, more recently, using artificial NNs.

In this chapter, we survey a number of broad applications where RBN recognition (and related works) are investigated using such phases. Various detection and recognition techniques discussed in literature are also detailed. We broadly define previous applications of artificial deep-learning NNs in these contexts.

2.1 Detection Strategies

Text extraction strategies have seen continuing interest in the literature, with many comprehensive surveys assessing the state of the art [12, 50, 51, 68, 119]. It is widely demonstrated that if text within an unstructured scene is *detected* reliably, then existing OCR engines can suitably extract these characters [100] once they exist in a structured context; thus not every extraction pipeline needs to self-contain a recognition strategy if commercial OCR packages suffice. A survey into the two prominent detection strategies is given in Sections 2.1.1 to 2.1.2.

These two prominent strategies have a varied nomenclature: (1) the CC-based (or *region*-based) approach, that utilise different region properties (e.g., colour, edges, CCs) [16, 26, 46, 54, 63, 65, 74, 75, 96, 97, 105, 106, 120, 121] for unsupervised extraction; and, (2) learning-based (or *texture*-based) approach, which uses unique texture properties to supervise extraction text from its background [18, 20, 23, 36, 38, 60, 64, 70, 82, 88, 89, 111, 116, 117]. Some authors have proposed methods that mix both supervised and unsupervised techniques [7, 79, 84].

2.1.1 CC-based techniques

CC-based approaches generate separated CCs using properties such as stroke width, pixel colour and edges, typically applying geometric and texture filters to reduce false positives. Neighbouring pixels are then 'grouped' using an algorithm originally presented by Horn [42].

Early works required the use scanning windows [18, 49, 70], limited by a constant image scale and discrete orientations of the sliding (thereby preventing text strokes in non-linear directions). Subramanian et al. [105] overcame this limitation by implementing a algorithm to detect text strokes by scanning an image horizontally and looking for sudden changes in background intensity (Figure 2.1b). However, this algorithm assumes a darker text on a lighter background to find such intensity changes, and consequently there are numerous parameters that must be fine-tuned. Additionally, the algorithm is only able to detect horizontal text only, and detected strokes are not grouped into characters, words and sentences.

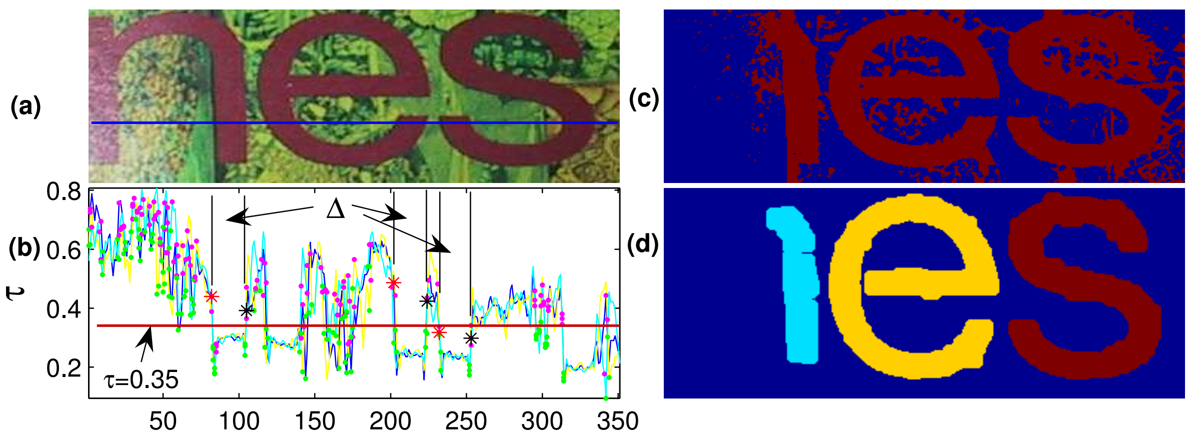

Figure 2.1: A study from Subramanian et al. [105] showed that stroke width could be determined from (a) the original image; (b) the intensity plots of the image to determine stroke regions; (c) the intensity at an optimal threshold; (d) the final thresholded image after morphological operations and CC analysis.

A study by Epshtein et al. [26] (and coincidentally Zhang and Kasturi [121]) built on the idea presented by Subramanian et al., introducing the Stroke Width Transformation (SWT) concept, a local image operator that determines the most likely stroke of a given pixel by computing the per-pixel width. This was later expanded in Srivastav and Kumar [104]. The SWT approach overcame previous limitations by introducing a system that can detect text regardless of size, typeface, direction and language, making it one of the first widely cited multilingual text detection algorithms. Additionally, SWT overcame methods that required the use of an OCR filtering stage to reduce false positives [18, 19, 117]. A sample of SWT is shown in Figure 2.2.

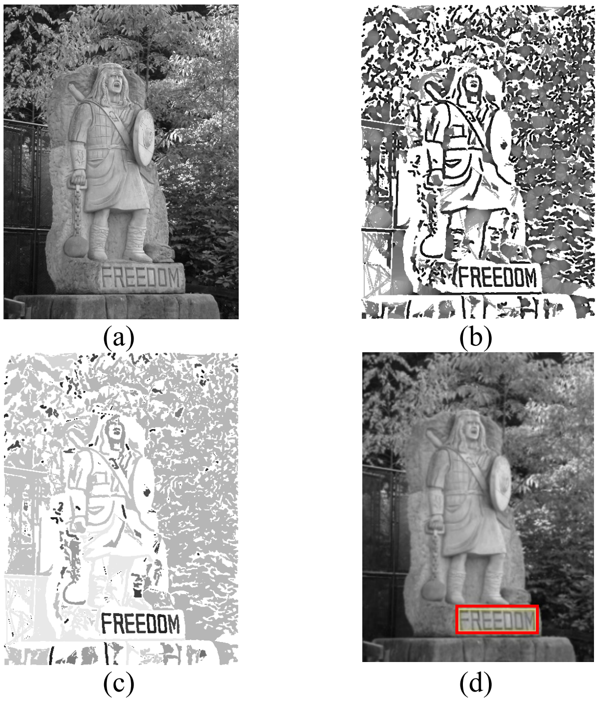

Figure 2.2: The Stroke Width Transformation (SWT) approach introduced in [26]. The original image (a) is converted to a binarised array with the most likely stroke width per-pixel (b), piping the information into geometric filtering (c) as text maintains fixed stroke width (excluding false positives such as foliage). The resulting detected text is shown in (d).

It is common to see edges computed from a raw image using the Canny-Edge Detection algorithm [10]. This was successfully applied in various CC-based studies [16, 26, 120]. While several papers have exploited SWT and adapted it further [97, 121], when opposite edges are not parallel, the SWT forms candidates with holes appearing in stroke curves or joints. This is due to candidates formed by shooting rays from detected the edges along the gradient found, removing the rays if terminated by another edge pixel of a perpendicular gradient. Further limitations include undetected stronger highlights, blurry text, and text with a wide curvature.

An alternate approach that overcomes this limitation was introduced by Chen et al. [16], where the complimentary properties of Canny-Edges [10] and Maximally Stable Extremal Regions (MSERs) [80] were combined. MSER is a detection mechanism suited for region-based detection, is robust against varying viewpoints, scales and illuminations [81], and can be extracted from images efficiently [86]. A limitation of MSER is its sensitivity to image blur [81], but Chen et al. demonstrated that MSER can be edge-enhanced using Canny-Edges on a contrast-enhanced image (Figure 2.3), achieving comparable results to SWT presented by Epshtein et al. [26]. Multiple works have utilised MSERs in a wide range of applications, such as their use in teaching CNNs and real-time text extraction [37, 45, 58, 65, 118].

Figure 2.3: Extracting text from a natural image shown in Chen et al. [16]. *From left to right:* Detected Maximally Stable Extremal Regions (MSERs) of black-on-white objects; text candidates grouped to formed text lines after geometric and stroke width filtering; false positives rejected using text verification showing detected text in the blue box.

A significant requirement of all CC-based techniques are the requirements to cluster extracted components back together again. This, in turn, also helps to remove any false positives by removing properties that don't meet set criteria. Various proposals have been made:

- Epshtein et al. [26] use basic geometric filtering based on the stroke width detected and height ratios of candidates. Additionally, colours of candidates are averaged as it is expected that words be written in the same colour. These are then clustered into candidates pairs (of at least three letters), chained together if they share a similar direction.

- Zhang and Kasturi [121] investigate the spacial relationship and property similarity of two neighbouring candidates, computing their link energies to compute *text energy* (the probability a candidate is a true positive). The distance of the text energies are computed and, where beyond a set threshold, will be eliminated if not met. This is presented in Figure 2.4.

- Zhang and Kasturi [120] expand the use of graph spectrum which has successfully been used in computer vision [92] to show image features in the form of a graph, use an adjacency matrix, then gather clusters of CCs based on the positive eigenvectors of the graph. This process is illustrated in Figure 2.5.

- Shivakumara et al. [97] propose the use of skeletal distance maps of a CC to remove small artificats and reduce false positives. They define *simple* and *complex* CCs respectively as: (a) a single text string or false positive; (b) multiple text strings which are connected to each other. The skeletonisation process is shown in Figure 2.6.

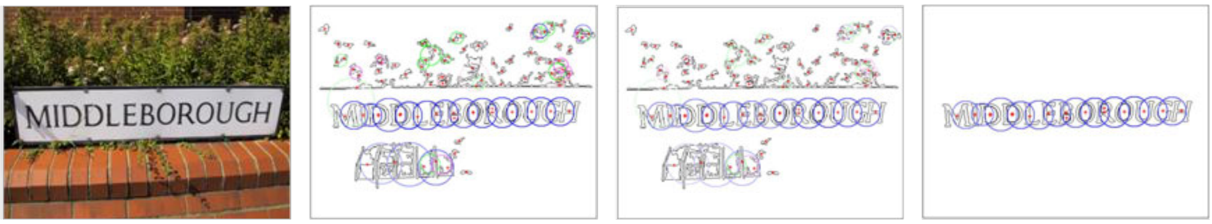

Figure 2.4: Using link and text energies for reconnecting character candidates shown in Zhang and Kasturi [121]. *From left to right:* original image; all link energies determined in a given image (note the false positives of background foliage); text energies calculated; all text energies greater than 0.5.

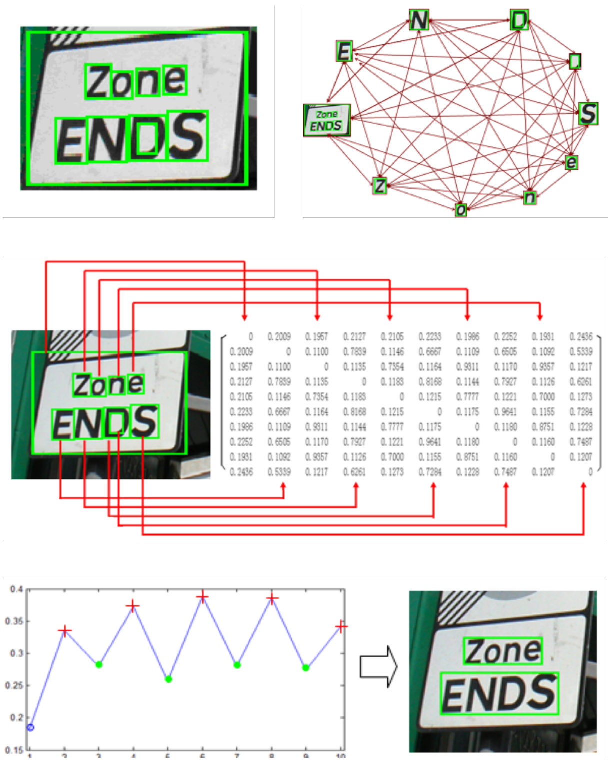

Figure 2.5: Process of grouping components via graph spectrum [120]. *Top-left:* 10 CCs detected. *Top-right:* generated graph from detected candidates. *Middle:* generated adjacency matrix. *Bottom-left:* Positive eigenvector resulting from the graph spectrum. *Bottom-right*: Resulting bounding boxes.

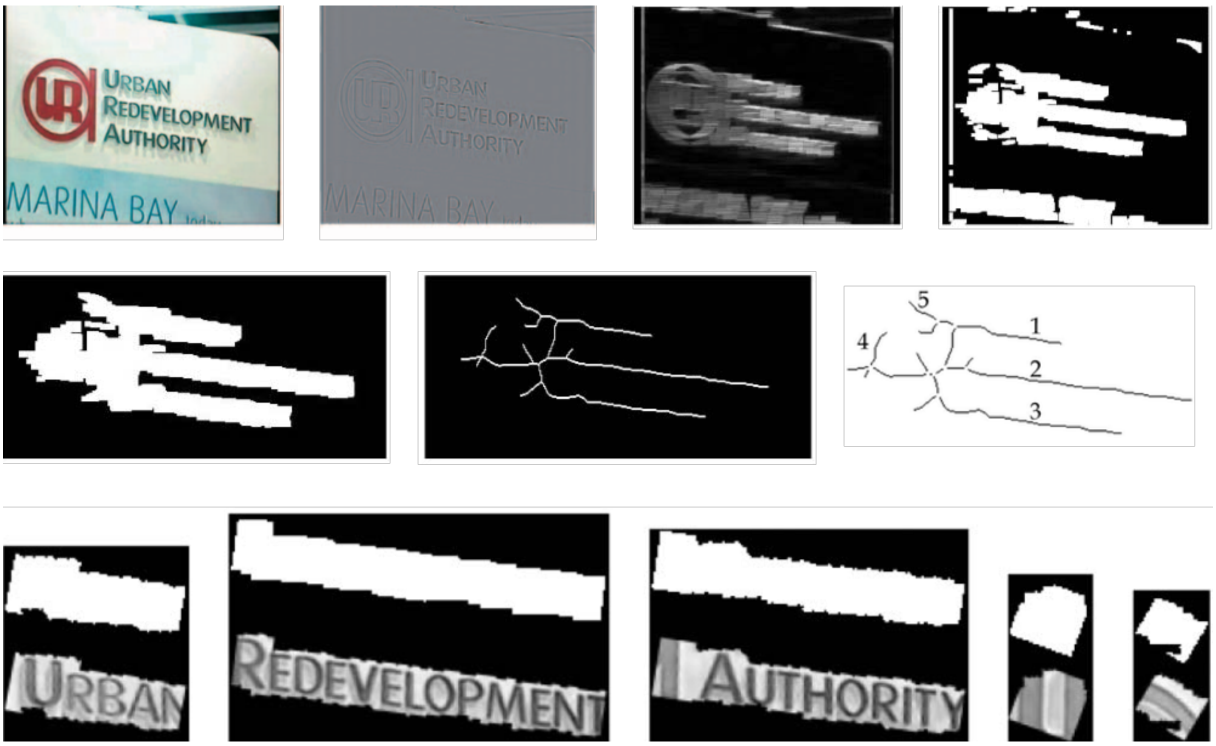

Figure 2.6: Developing a skeletal map using the process proposed by [97]. *Top row:* Original image is processed using Fourier-Laplacian filtering. A maximum distance map is developed and parsed through a morphological operation to remove smaller artefacts. *Middle row:* CC classification and further skeletonisation, showing five labelled subcomponents. *Bottom row:* The five sample subcomponents extracted from the skeletonisation process (in order). Note that subcomponents 4 and 5 are false positives.

2.1.2 Learning-based techniques

The learning-based approach has had a varied popularity over the years. While also referred to as a *texture*-based approach, it typically utilises learning-based methods to train a classifier using these textures, considering text as a special texture within the image. This is done by extracting certain features over a portion of the image (i.e., the texture) either via heuristics or machine learning. The texture is typically extracted by scanning at varying scales, then classifying areas of pixels based on certain features. A classifier uses these features to identify text from non-text to extract text from its background.

Early works in the area focused primarily how to classify a region as either text or non-text [55, 64, 70, 102]. However, it was reported that, while it may be easy to procure training samples of text, it is often more difficult to procure non-text training samples [41, 107] due to the wide variance of what 'non-text' samples actually are, and ensuring they are well-represented in the training set of the classifier.

A varied range of features have been utilised to mark textures. A popular one is the Histogram of Oriented Gradient (HOG), a object detection technique that utilises gradient orientation, first conceptualised in [1] and later coined by Freeman and Roth [28]. However, its popularity did not become widespread until [23], where Dalal and Triggs showed its applicability on pedestrian detection in natural photos. Later, HOG was shown to be useful in text detection by Hanif et al. [39]. Further features include: Local Binary Patterns (LBP) [87], which was shown to improve detection when combined with HOG [115]; wavelet energy [83, 113], which were applied to extract subtitles [36] and worked optimally with HOG in text extraction [88] (when compared to other features); Gabor filters [72]; Scale Invariant Feature Transform (SIFT) descriptors [76]; grey scale features [55]; Speeded-Up Robust Features (SURF) [4], edge map features [13], and shape contexts [5].

Many of these features can be combined and fed into a single or multiple classifiers [36, 38, 39, 88, 111, 116, 117]. Example classifiers include Support Vector Machines (SVMs) [9, 22, 112], the Adaptive Boosting (AdaBoost) classifier [29]—and variants thereof ([30, 38, 103])—or NNs, such as Multilayer Perceptrons (MLPs). Chen et al. [14] report that the use of SVMs showed better text texture verification than MLPs.

As shown in the wide range of features and classifiers, a main limitation of learning-based methods is the difficulty in selecting which combinations of features to use, the inability to detect sufficiently slanted text, as well as its reported high computational complexity due to the need to scan the image at different scales [26, 65]. Furthermore, it is typical for these classifiers to required thousands of training images [15].

More recently, deep-learning CNNs have been utilised for instance classification and per-pixel segmentation, as emphasised in Figure 2.8. While CNNs are a relatively old concept (see LeCun et al. [59]), they lost interest within image processing to Support Vector Machines (SVMs) and AdaBoost throughout the late 1990s and 2000s. In 2012, however, Krizhevsky et al. [56] rekindled interest by demonstrating far increased classification accuracy within images in the ImageNet Large Scale Visualisation Recognition Challenge [24], with only a few modifications made to the CNN proposed by LeCun et al. more than a decade prior.

This has since led to the rise of enhanced CNNs such as FICS+++ [66], R-CNN [34], Fast/Faster R-CNN [35, 90], and Mask R-CNN [40]. A comparison of two of these networks using images in the Microsoft Common Objects in Context [71] dataset are shown in Figure 2.8, with the Mask R-CNN framework illustrated in Figure 2.7. The rise of these recent developments have sparked interest in multi-language machine learning libraries, such as MXNet[1], which are able to combine these academic works for use in large-scale industry-focused production code [17].

These methods are increasingly improving per-pixel classification of objects within natural scenes. However, applicability of these deep-learning networks in the sole context of *text extraction* is yet to be widely explored.

[1] `http://mxnet.io` last accessed 30 June 2017.

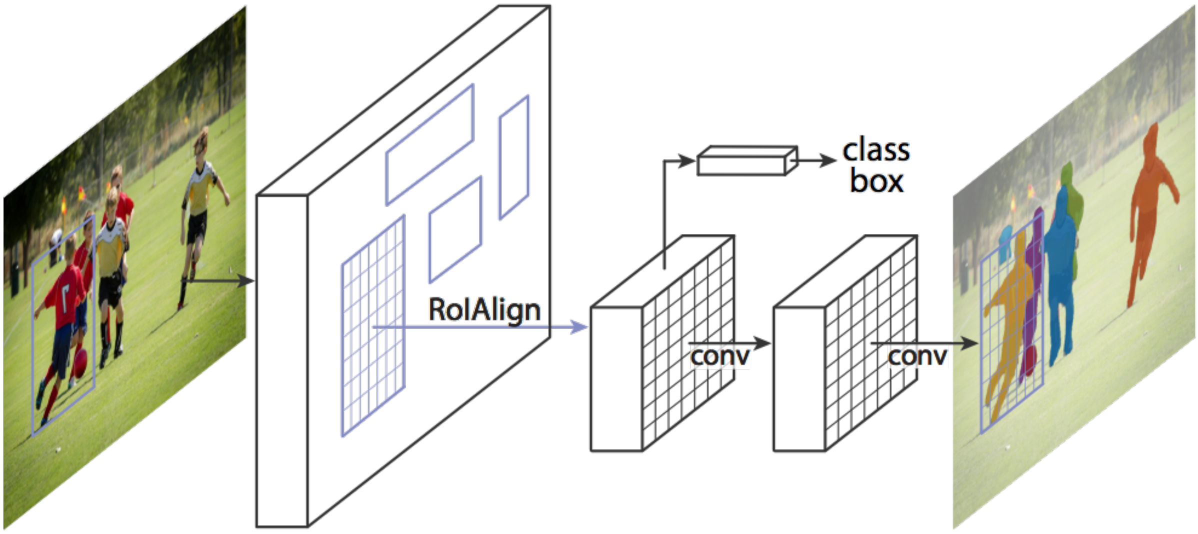

Figure 2.7: The Mask R-CNN framework for instance segmentation [40].

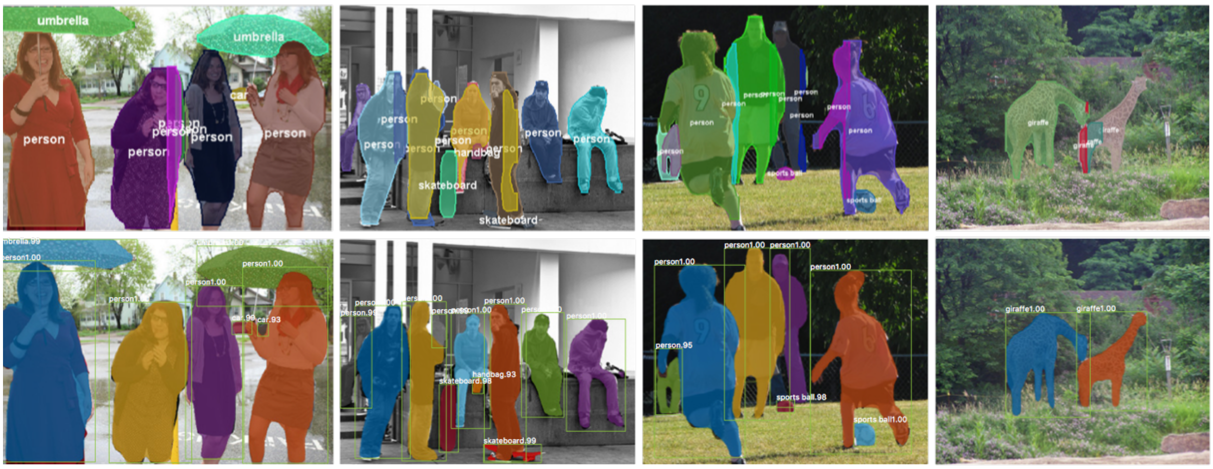

Figure 2.8: Use of CNNs have recently been shown to have accurate per-pixel object detection and classification. *Top row:* FCIS+++ [66]. *Bottom row:* Mask R-CNN [40]. Mask R-CNN outperforms FICIS+++ for overlapping instance segmentation.

2.2 Recognition Strategies

The International Conference on Document Analysis and Recognition (ICDAR) Robust Reading Competitions [52, 53, 77, 78, 95] broke down the issue of text extraction into two sub-problems: text locating and character recognition. Most of the literature discussed in Section 2.1 focused within the text locating sub-problem. For character recogntion, the *Focused Scene Text* word recognition task[2] received three entries in 2011 and four in 2013. In 2015, the recognition challenge was redesigned[3] using a new (and more challenging) *Incidental Scene Text* dataset of photos captured in-the-wild using Google Glass. The evaluation scheme uses the *Total Edit Distance* metric (described in [53]) and additionally the number of correctly recognised words for qualitative analysis. Table 2.1 summarises the top-scoring recognition rates from these competitions.

Table 2.1: Top-scoring word recognition results from ICDAR 2011–2015.

Year	**Method**	**Dataset**	**Total Edit Distance**	**Correctly Recognised Words (%)**
2011	TH-OCR System [73]	Focused	176.23	41.2
2013	PhotoOCR [8]	Focused	122.7	82.83
2015	MAPS [57]	Incidental	1128.0	32.93

It has been widely demonstrated that off-the-shelf commercial and open source OCR packages are able to correctly recognise text once the characters are extracted. In their conclusions of the ICDAR 2015 competition, Karatzas et al. note that the top performing methods will make use of commercial OCRs and conclude that conventional shape-based OCR engines can produce competitive results with pre-processed images.

This conclusion is emphasised in further works. The Open Source Tesseract OCR engine was used in Ben-ami et al. [6] to extract RBNs after preprocessed CC-based extraction. Similarly, leading commercial OCR engines used by Chen and Yuille [18] were able to achieve 93% recognition from binarised text regions after AdaBoost non-text classification, using ABBYY FineReader[4], TOCR[5] and Readiris Pro[6]. Similarly, Gatos et al. [32] used ABBYY FineReader and showed their extraction method showed a 50% improvement for indoor and outdoor scene images, an approach also applied by Chen and Yuille [18].

[2]See Challenge 2, Task 3 in [53, 95].
[3]See Challenge 4, Task 3 in [52].
[4]`https://www.abbyy.com` last accessed 3 July 2017.
[5]`http://www.transym.com` last accessed 3 July 2017.
[6]`http://www.irislink.com/readiris` last accessed 3 July 2017.

This is not to say that OCR engines are always needed: interest in developing novel general character recognition strategies have also been investigated. Wang et al. [114] compared ABBYY FineReader with their novel PLEX approach and demonstrated that their text recognition system can outperform traditional OCR engines without the use of a text detector. This inspired more recent works, such as the use of lexicon-free photo OCR frameworks by Lee and Osindero [61]. Furthermore, application-specific recognition has been investigated using variant techniques.

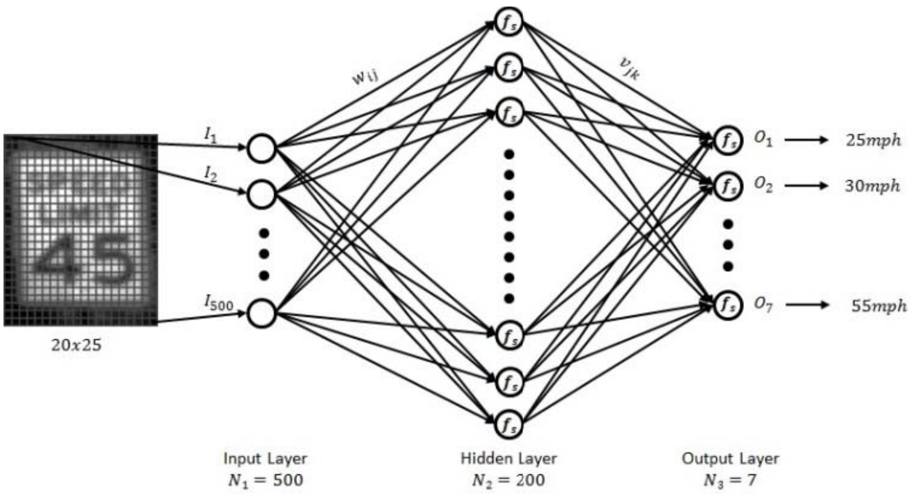

Figure 2.9: The artificial Multilayer Perceptron (MLP) NN designed in Kundu and Mackens [58] to recognise US-style speed limit signs.

A 2015 study into Traffic Sign Recognition (TSR) to detect US-style speed limit signs achieved recognition without the use of any OCR packages. In [58], Kundu and Mackens extracted a speed limit sign via the use of MSERs and template matching. The resulting detected signs were scaled to a grayscaled size of 20×25 pixels and fed into a Multilayer Perceptron (MLP) NN of 200 neurons in the hidden layer. The output layer of the network consisted of seven nodes, each representing the seven kinds of speed limit signs in US cities (25, 30, 35, 40, 45, 50, and 55 miles per hour). This architecture is shown in Figure 2.9. When trained with 13,289 images of text cases and 4,319 non-text cases, the results showed that their recognition classier was able to correctly recognise speed limit signs with an accuracy of 98.04%. Similar results were achieved using a feed-forward MLP by Eichner and Breckon [25], using UK/Poland style speed limits scaled to 20×20 pixels (grayscale) and 12 output layer neurons (10–100 kilometres per hour, the national speed limit sign, and non-sign neurons).

However, works in TSR systems that utilise NNs are generally non-generalisable, and only work in a limited context (i.e., by classifying speed limit signs of known outputs). In our context of RBN recognition, we have a known character output range of 36 possibilities (0–9 and A–Z = 10 + 26).

Beyond TSR systems, however, we see the use of more generalisable networks: Netzer et al. [85] trained neural network to recognise street number characters from Google Street View with higher precision and recall than that of HOG and the Tesseract OCR engine, showing that the applicability of NNs for recognition can outperform traditional means. Anagnostopoulos et al. [2] used a Probabilistic Neural Network (PNN) to recognise single characters of the same 36 possibility range (i.e., uppercase alphanumeric characters) corresponding to the input grayscale vector of 9×12 pixels ($9 \times 12 = 108$ input neurons) for a single character. Figure 2.10 illustrates the PNN architecture used in this study. Furthermore, investigations in comparing different architectures of NNs for this context is given in Lee and Osindero [61].

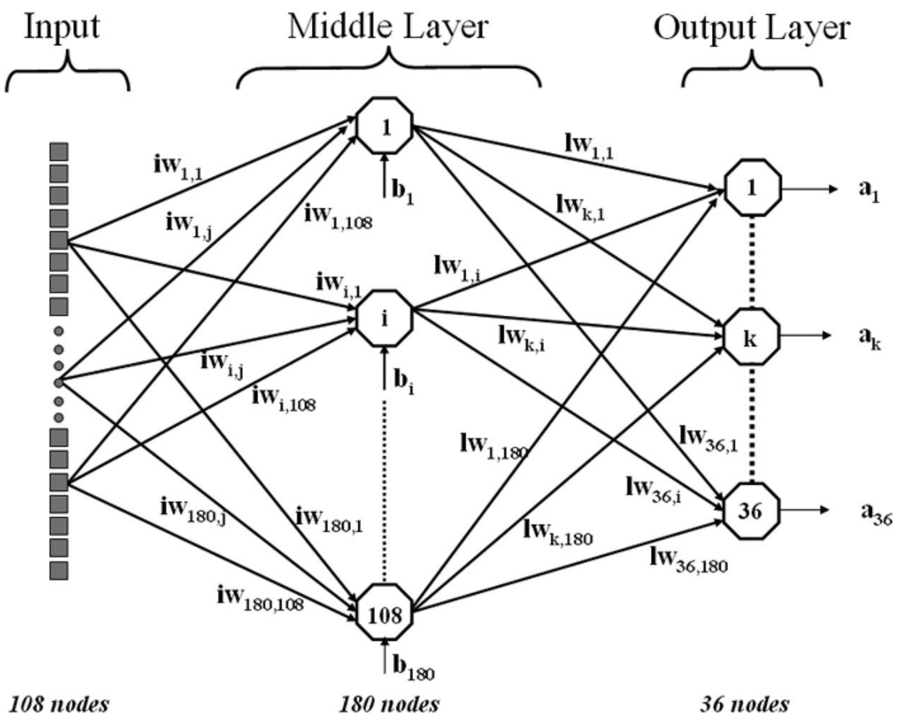

Figure 2.10: Anagnostopoulos et al. [2] developed the architecture of a PNN to determine a single character from within a license plate.

However, these pipelines are still entirely dependent on good detection strategies, and in the case where classifiers are used, quality training data must be supplied. The pipeline developed by Ben-ami et al. [6] for RBN detection (Figure 2.11) is dependent on quality facial detection: the OpenCV implementation [69] was used in this study. In order to detect the torso region—and thus detect the bib sheet itself—a heuristically-driven calculation is used to hypothesise where the torso bounds ($T_b = T_h \times T_w$) are, given by the face height (F_h), face width (F_w):

$$T_b = (3 \times F_h) \times (\frac{7}{3} F_w)$$

The location of T_b is horizontally located at the centre of the face midpoint and vertically halfway below the face height ($0.5 \times F_w$). Hence, the dependency on these heuristics are heavily driven by a tolerant bounding box, which in itself depends on the face detection itself. As Fu et al. [31] have also noted, even if the face is detected, the bib sheet can be obfuscated (e.g., by another runner) or if the camera is capturing on the runner's side, the OCR engine or face-detection algorithm may produce poor results. In our study, we ignore these limitations for the intention of capturing prominent runners.

Furthermore, the use of Tesseract in the study meant significant filtering, separation and character alignment is needed for the OCR engine to read characters correctly (Figure 2.12). This is yet another step which can possibly be avoided via the use of well-trained NNs, as demonstrated in previous works. Investigation into applying such networks in this context (i.e., alphanumeric sequences on *human* subjects) is largely lacking.

Figure 2.11: The RBN recognition pipeline by Ben-ami et al. [6]. *From left to right:* Input image; face detection results in red and the estimated RBN region hypothesis in green; SWT of the hypothesis region; tag region detection in blue; tag region after digit processing.

Figure 2.12: Character processing in [6]. *From left to right:* A detected tag; separation via SWT CC analysis; binarised characters; CC analysis and filtering; separation and alignment.

2.3 Metrics

Throughout our survey, we have utilised the evaluation scheme first proposed for use in image processing in the International Conference on Document Analysis and Recognition (ICDAR) competitions [52, 53, 77, 78, 95]. This scheme was designed to be easy to understand and compute, reward text extraction useful for natural scenes, and heavily punish trivial solutions. The intention behind these metrics were to develop a measure of 'robustness' a text extraction pipeline can achieve.

2.3.1 Precision and Recall

Generally in information retrieval, the precision (p) and recall (r) metrics are used, first defined in the six evaluation criteria for information retrieval systems by Cleverdon et al. [21]. Precision refers to the proportion of relevant matches actually retrieved in the retrieved results, while recall refers to the proportion of relevant matches retrieved in total relevant instances. We use recall and precision metrics to assess the *effectiveness* of an information retrieval system [91].

In the context of image processing, systems that over-estimate are punished with a low precision score, while systems that under-estimate are punished with a low recall score [78]. Therefore, precision is the number of correct candidates (c) divided by the number of total estimates found (E):

$$p = \frac{c}{|E|}$$

And recall is defined as the number of correct estimates divided by the total number of ground-set truth targets (T):

$$r = \frac{c}{|T|}$$

However, it is not realistic for a given text extraction pipeline to *exactly* agree with the rectangle bounds manually tagged by a human. Lucas et al. [78] proposed changes to these calculations to better suit their usage in the context of information extraction from within images. They adopt a more flexible notion of what a 'match' is. They define a new match measure (m_p) between two rectangles (i.e., the ground truth and the system's detected candidate) as "the area of intersection of both rectangles divided by the area of the minimum bounding box containing both rectangles" [78]. This allows for a match value of one when the candidate is identical to the ground truth, and zero where the candidate has no intersection at all to the ground truth.

Therefore, the best match, $m(r, R)$, of a rectangle r in a set of rectangles R is:

$$m(r, R) = \max m_p(r, r') \mid r' \in R$$

Lastly, we can redefine the recall and precision metrics to be more forgiving in the image extraction context:

$$p' = \frac{\sum_{r_e \in E} m(r_e, T)}{\mid E \mid}$$

$$r' = \frac{\sum_{r_t \in T} m(r_t, E)}{\mid T \mid}$$

2.3.2 The f-score

Common metrics used when developing text extraction pipelines utilise the use of the f-score, a single measure of quality that combines both precision and recall values computed above. We are able to compute this metric using the standard measure across many studies, as contrasted in Table 2.2.

The f-score algorithm is given in the context of image processing in Lucas et al. [78]. Relative weights controlled by an α value of 0.5 give equal weight to both precision and recall metrics:

$$f = \frac{1}{\frac{\alpha}{p'} + \frac{1-\alpha}{r'}}$$

As Chen et al. [16] report, even when all text is correctly localised, it is likely that the f-score will vary between 0.8–1.0. This is because E boundaries are unlikely to match *exactly* with the manually labelled T boundaries. An illustration of this is phenomena is shown in Figure 2.13.

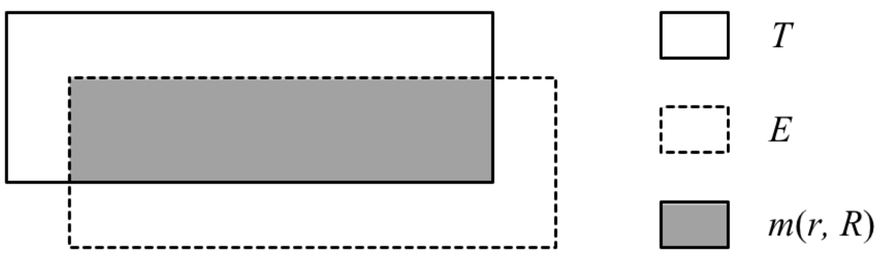

Figure 2.13: Overlapping areas of the ground truth targets, T, the estimated target boundaries E and the best match $m(r, R)$. (Adapted from [117].)

Table 2.2: A survey of text extraction literature, separated into CC- and learning-based detection methods.

Reference	Scientific Background		Precision	Recall	f-score	Dataset(s)	Performance	
	Detection	**Recognition**					**Platform**	**Time (s)**
[6]	*CC-based*: SWT; Canny-Edges; Binary Conversion; Geometric Filtering; CC-Alignment	OCR (Tesseract)	0.65	0.62	0.63	N/A	N/S	N/S
[16]	*CC-based*: Canny-Edges; MSER; SWT, Geometric, Template Matching	OCR (N/S†)	0.73	0.60	0.66	[77, 78]	2.5 GHz	0.20
[65]	*CC-based*: MSER; geometric and SWT filters; skeletal distance mapping	N/S	0.59 [78] 0.59 [95]	0.59 [78] 0.62 [95]	0.59 [78] 0.61 [95]	[77, 95]	N/S	N/S
[121]	*CC-based*: character detection with HOG and SWT; link energies via spacial relationships between characters	N/S	0.73	0.62	0.67	[78]	N/S	N/S
[97]	*CC-based*: Fourier-Laplacian Filtering; clustering based on maximum distance via K-Means; skeletal distance mapping	N/S	0.76 [78] 0.81 [44]	0.86 [78] 0.93 [44]	0.81 [78] 0.87 [44]	[44, 78]	2.0 GHz	7.80
[26]	*CC-based*: Canny-Edges; SWT; Modified CC algorithm; Geometric Filtering	OCR (N/S)	0.73	0.60	0.66	[77, 78]	N/S	0.94
[120]	*CC-based*: Canny-Edges; HOG; geometric filtering; graph spectrum	N/S	0.67	0.46	0.55	[78]	N/S	N/S
[117]	*Learning-based*: SVM to reject non-text; wavelet movement; HOG; OCR filtering	N/S	N/S	0.97	N/S	[44]	1.6 GHz	8.30
[88]	*Learning-based*: Waldboost Classifier; HOG; LBP; Gabor Wavelets	N/S	0.56	0.70	0.68	[78]	3.4 GHz	0.37
[36]	*Learning-based*: Wavelet transformation; feature estimation; pixel-block classification	N/S	0.87	0.90	0.88	[43]	N/S	N/S
[38]	*Learning-based*: Mean Difference, Standard Deviation and HOG features; AdaBoost and CAdaBoost classifiers; NN-baed localiser	N/S	0.25	0.35	0.35	[78]	2.2 GHz	N/S

†Not Specified

Summary

In this chapter, we have surveyed a range of literature in various application contexts: RBN and TSR recognition, recognition of alphanumeric sequences ‘in the wild’, and additionally object instance segmentation. We have also investigated the varied range of techniques used to both detect and recognise the text, using both heuristic-based and NN-based approaches. Our following chapter will discusses a means to capture data for the purposes of training a deep-learning NN.

Chapter 3

Data Set

Chapter 4

Benchmarking

4.1 Open Source Tools

4.2 Existing Pipelines From Literature

4.3 Hermes Approach

Chapter 5

Processing Pipeline

Chapter 6

Findings

Chapter 7

Discussion

Chapter 8

Conclusions and Future Work

References

[1] Method of and apparatus for pattern recognition.

[2] Anagnostopoulos, C.-N., I. Anagnostopoulos, V. Loumos, and E. Kayafas (2006). A License Plate-Recognition Algorithm for Intelligent Transportation System Applications. *IEEE Trans. Intelligent Transportation Systems*.

[3] Anagnostopoulos, C.-N., I. Anagnostopoulos, I. D. Psoroulas, V. Loumos, and E. Kayafas (2008). License Plate Recognition From Still Images and Video Sequences - A Survey. *IEEE Trans. Intelligent Transportation Systems*.

[4] Bay, H., A. Ess, T. Tuytelaars, and L. J. Van Gool (2008). Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*.

[5] Belongie, S. J., J. Malik, and J. Puzicha (2001). Matching Shapes. *ICCV*.

[6] Ben-ami, I., T. Basha, and S. Avidan (2012). Racing Bib Numbers Recognition. In *British Machine Vision Conference 2012*.

[7] Bengio, Y., P. Lamblin, D. Popovici, and H. Larochelle (2006). Greedy Layer-Wise Training of Deep Networks. *NIPS*.

[8] Bissacco, A., M. Cummins, Y. Netzer, and H. Neven (2013). PhotoOCR - Reading Text in Uncontrolled Conditions. *ICCV*.

[9] Burges, C. J. C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Min. Knowl. Discov.*.

[10] Canny, J. F. (1986). A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.*.

[11] Cano-Perez, J. and J. C. Pérez-Cortes (2003). Vehicle License Plate Segmentation in Natural Images. *IbPRIA*.

[12] Chen, D. and J. Luettin (2000). A survey of text detection and recognition in images and videos.

[13] Chen, D., J.-M. Odobez, and H. Bourlard (2004a). Text detection, recognition in images and video frames. *Pattern Recognition*.

[14] Chen, D., J.-M. Odobez, and H. Bourlard (2004b). Text detection, recognition in images and video frames. *Pattern Recognition*.

[15] Chen, D., J.-M. Odobez, and J.-P. Thiran (2004). A localization/verification scheme for finding text in images and video frames based on contrast independent features and machine learning methods. *Sig. Proc. - Image Comm.*.

[16] Chen, H., S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk, and B. Girod (2011). Robust text detection in natural images with edge-enhanced Maximally Stable Extremal Regions. *ICIP*.

[17] Chen, T., M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang (2015). MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems. *arXiv.org*.

[18] Chen, X. and A. L. Yuille (2004a). Detecting and reading text in natural scenes. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004*.

[19] Chen, X. and A. L. Yuille (2004b). Detecting and Reading Text in Natural Scenes. *CVPR*.

[20] Chen, X. and A. L. Yuille (2005). A Time-Efficient Cascade for Real-Time Object Detection - With applications for the visually impaired. *CVPR Workshops*.

[21] Cleverdon, C., J. Mills, and M. Keen (1966). Factors Determining the Performance of Indexing Systems. Technical report, ASLIB Cranfield Research Project.

[22] Cortes, C. and V. Vapnik (1995). Support-Vector Networks. *Machine Learning*.

[23] Dalal, N. and B. Triggs (2005). Histograms of Oriented Gradients for Human Detection. *CVPR*.

[24] Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009). ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[25] Eichner, M. L. and T. P. Breckon (2008). Integrated speed limit detection and recognition from real-time video. In *2008 IEEE Intelligent Vehicles Symposium (IV)*.

[26] Epshtein, B., E. Ofek, and Y. Wexler (2010). Detecting text in natural scenes with stroke width transform. *CVPR*.

[27] Faloutsos, C., R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz (1994). Efficient and effective Querying by Image Content. *Journal of Intelligent Information Systems*.

[28] Freeman, W. T. and M. Roth (1995). Orientation histograms for hand gesture recognition. *International workshop on automatic ...*.

[29] Freund, Y. and R. E. Schapire (1996). Experiments with a New Boosting Algorithm. *ICML*.

[30] Friedman, J., R. Tibshirani, and T. Hastie (2000). Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors). *The annals of statistics*.

[31] Fu, C., C.-W. Cheng, W.-H. Shen, Y.-L. Wei, and H.-M. Tsai (2015). LightBib: Marathoner Recognition System with Visible Light Communications. In *2015 IEEE International Conference on Data Science and Data Intensive Systems (DSDIS)*.

[32] Gatos, B., I. Pratikakis, and K. Kepene (2005). Text detection in indoor/outdoor scene images. *Proc First Workshop of ...*.

[33] Girod, B., V. Chandrasekhar, D. Chen, N.-M. Cheung, R. Grzeszczuk, Y. Reznik, G. Takacs, S. Tsai, and R. Vedantham (2011). Mobile Visual Search. *IEEE Signal Processing Magazine*.

[34] Girshick, R., J. Donahue, T. Darrell, and J. Malik (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[35] Girshick, R. B. (2015). Fast R-CNN. *ICCV*.

[36] Gllavata, J., R. Ewerth, and B. Freisleben (2004). Text Detection in Images Based on Unsupervised Classification of High-Frequency Wavelet Coefficients. *ICPR*.

[37] Gonzalez, Á., L. M. Bergasa, J. J. Y. Torres, and S. Bronte (2012). Text location in complex images. *ICPR*.

[38] Hanif, S. M. and L. Prevost (2009). Text Detection and Localization in Complex Scene Images using Constrained AdaBoost Algorithm. *ICDAR*.

[39] Hanif, S. M., L. Prevost, and P. Negri (2008). A cascade detector for text detection in natural scene images. *ICPR*.

[40] He, K., G. Gkioxari, P. Dollár, and R. B. Girshick (2017). Mask R-CNN. *CoRR*.

[41] Heisele, B., T. Serre, S. Mukherjee, and T. A. Poggio (2001). Feature Reduction and Hierarchy of Classifiers for Fast Object Detection in Video Images. *CVPR*.

[42] Horn, B. (1986). *Robot Vision*. MIT Press.

[43] Hua, X.-S., L. Wenyin, and H. Zhang (2001). Automatic Performance Evaluation for Video Text Detection. *ICDAR*.

[44] Hua, X.-S., L. Wenyin, and H. Zhang (2004). An automatic performance evaluation protocol for video text detection algorithms. *IEEE Trans. Circuits Syst. Video Techn.*.

[45] Huang, X., T. Shen, R. Wang, and C. Gao (2015). Text detection and recognition in natural scene images. In *2015 International Conference on Estimation, Detection and Information Fusion (ICEDIF)*.

[46] Jain, A. K. and B. Y. 0002 (1998). Automatic text location in images and video frames. *ICPR*.

[47] Jin, C. M., Z. Omar, and M. H. Jaward (2016). A mobile application of American sign language translation via image processing algorithms. In *2016 IEEE Region 10 Symposium (TENSYMP)*.

[48] Jin, J., K. Fu, and C. Zhang (2014). Traffic Sign Recognition With Hinge Loss Trained Convolutional Neural Networks. *IEEE Transactions on Intelligent Transportation Systems*.

[49] Jung, C., Q. Liu, and J. Kim (2009). A stroke filter and its application to text localization. *Pattern Recognition Letters*.

[50] Jung, K., K. In Kim, and A. K Jain (2004). Text information extraction in images and video: a survey. *Pattern Recognition*.

[51] Jung, K., K. I. Kim, and A. K. Jain (2004). Text information extraction in images and video - a survey. *Pattern Recognition*.

[52] Karatzas, D., L. Gomez-Bigorda, A. Nicolaou, S. K. Ghosh, A. D. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny (2015). ICDAR 2015 competition on Robust Reading. *ICDAR*.

[53] Karatzas, D., F. Shafait, S. Uchida, M. Iwamura, L. G. i. Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. de las Heras (2013). ICDAR 2013 Robust Reading Competition. In *2013 12th International Conference on Document Analysis and Recognition (ICDAR)*.

[54] Kim, H.-K. (1996). Efficient Automatic Text Location Method and Content-Based Indexing and Structuring of Video Database. *J. Visual Communication and Image Representation*.

[55] Kim, K. I., K. Jung, and J. H. Kim (2003). Texture-Based Approach for Text Detection in Images Using Support Vector Machines and Continuously Adaptive Mean Shift Algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*.

[56] Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). ImageNet Classification with Deep Convolutional Neural Networks. *NIPS*.

[57] Kumar, D., M. N. A. Prasad, and A. G. Ramakrishnan (2012). *MAPS: midline analysis and propagation of segmentation*. ACM.

[58] Kundu, S. K. and P. Mackens (2015). Speed Limit Sign Recognition Using MSER and Artificial Neural Networks. *ITSC*.

[59] LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*.

[60] Lee, C. W., K. Jung, and H. J. Kim (2003). Automatic text detection and removal in video sequences. *Pattern Recognition Letters*.

[61] Lee, C.-Y. and S. Osindero (2016). Recursive Recurrent Nets with Attention Modeling for OCR in the Wild.

[62] Lee, E. R., P. K. Kim, and H. J. Kim (1994). Automatic Recognition of a Car License Plate using Color Image Processing. *ICIP*.

[63] Lee, S., M. S. Cho, K. Jung, and J. H. Kim (2010). Scene Text Extraction with Edge Constraint and Text Collinearity. *ICPR*.

[64] Li, H., D. S. Doermann, and O. E. Kia (2000). Automatic text detection and tracking in digital video. *IEEE Trans. Image Processing*.

[65] Li, Y. and H. Lu (2012). Scene text detection via stroke width. *ICPR*.

[66] Li, Y., H. Qi, J. Dai, X. Ji, and Y. Wei (2016). Fully convolutional instance-aware semantic segmentation. *arXiv.org*.

[67] Lian, Z., X. Jing, S. Sun, and H. Huang (2016). Frequency Selective Convolutional Neural Networks for Traffic Sign Recognition. In *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring*.

[68] Liang, J., D. S. Doermann, and H. Li (2005). Camera-based analysis of text and documents - a survey. *IJDAR*.

[69] Lienhart, R. and J. Maydt (2002). An extended set of Haar-like features for rapid object detection. *ICIP*.

[70] Lienhart, R. and A. Wernicke (2002). Localizing and segmenting text in images and videos. *IEEE Trans. Circuits Syst. Video Techn.*.

[71] Lin, T.-Y., M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick (2014). Microsoft COCO - Common Objects in Context. *ECCV*.

[72] Liu, C. L., M. Koga, and H. Fujisawa (2005). Gabor feature extraction for character recognition: comparison with gradient feature. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*.

[73] Liu, H. and X. Ding (2005). Handwritten Character Recognition Using Gradient Feature and Quadratic Classifier with Multiple Discrimination Schemes. *ICDAR*.

[74] Liu, Y., S. Goto, and T. Ikenaga (2006). A Contour-Based Robust Algorithm for Text Detection in Color Images. *IEICE Transactions*.

[75] Liu, Z. and S. Sarkar (2008). Robust outdoor text detection using text intensity and shape features. *ICPR*.

[76] Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*.

[77] Lucas, S. M. (2005). ICDAR 2005 text locating competition results. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*.

[78] Lucas, S. M., A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young (2003). ICDAR 2003 robust reading competitions. In *Seventh International Conference on Document Analysis and Recognition*.

[79] Mairal, J., F. R. Bach, J. Ponce, G. Sapiro, and A. Zisserman (2008). Discriminative learned dictionaries for local image analysis. *CVPR*.

[80] Matas, J., O. Chum, M. Urban, and T. Pajdla (2002). Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. *BMVC*.

[81] Mikolajczyk, K., T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. J. Van Gool (2005). A Comparison of Affine Region Detectors. *International Journal of Computer Vision*.

[82] Minetto, R., N. Thome, M. Cord, J. Fabrizio, and B. Marcotegui (2010). Snoopertext - A multiresolution system for text detection in complex visual scenes. *ICIP*.

[83] Mohan, A., C. Papageorgiou, and T. A. Poggio (2001). Example-Based Object Detection in Images by Components. *IEEE Trans. Pattern Anal. Mach. Intell.*.

[84] Mutch, J. and D. G. Lowe (2006). Multiclass Object Recognition with Sparse, Localized Features. *CVPR*.

[85] Netzer, Y., T. Wang, and A. Coates (2011). Reading digits in natural images with unsupervised feature learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*.

[86] Nistér, D. and H. Stewénius (2008). Linear Time Maximally Stable Extremal Regions. *ECCV*.

[87] Ojala, T., M. Pietikainen, and D. Harwood (1994). Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In *12th International Conference on Pattern Recognition*.

[88] Pan, Y.-F., C.-L. Liu, and X. Hou (2010). Fast scene text localization by learning-based filtering and verification. In *2010 17th IEEE International Conference on Image Processing (ICIP 2010)*.

[89] Phan, T. Q., P. Shivakumara, and C. L. Tan (2009). A Laplacian Method for Video Text Detection. In *2009 10th International Conference on Document Analysis and Recognition*.

[90] Ren, S., K. He, R. B. Girshick, and J. S. 0001 (2017). Faster R-CNN - Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.*.

[91] Rijsbergen, C. J. V. (1979). *Information Retrieval*. Butterworth-Heinemann.

[92] Sarkar, S. and K. L. Boyer (1996). Quantitative Measures of Change based on Feature Organization - Eigenvalues and Eigenvectors. *CVPR*.

[93] Seo, Y.-W., J. Lee, W. Zhang, and D. Wettergreen (2015). Recognition of Highway Workzones for Reliable Autonomous Driving. *IEEE Transactions on Intelligent Transportation Systems*.

[94] Sermanet, P. and Y. LeCun (2011). Traffic sign recognition with multi-scale Convolutional Networks. *IJCNN*.

[95] Shahab, A., F. Shafait, and A. Dengel (2011). ICDAR 2011 Robust Reading Competition Challenge 2: Reading Text in Scene Images. In *2011 International Conference on Document Analysis and Recognition (ICDAR)*.

[96] Shivakumara, P., W. Huang, T. Q. Phan, and C. L. Tan (2010). Accurate video text detection through classification of low and high contrast images. *Pattern Recognition*.

[97] Shivakumara, P., T. Q. Phan, and C. L. Tan (2011). A Laplacian Approach to Multi-Oriented Text Detection in Video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[98] Sivic, J. and A. Zisserman (2003). Video Google - A Text Retrieval Approach to Object Matching in Videos. *ICCV*.

[99] Smeulders, A. W. M., M. Worring, S. Santini, A. Gupta, and R. C. Jain (2000). Content-Based Image Retrieval at the End of the Early Years. *IEEE Trans. Pattern Anal. Mach. Intell.*.

[100] Smith, R. (2007). An Overview of the Tesseract OCR Engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2*.

[101] Smith, R. W. (1987). *The Extraction and Recognition of Text from Multimedia Document Images*. Ph. D. thesis.

[102] Sobottka, K., H. Bunke, and H. Kronenberg (1999). Identification of Text on Colored Book and Journal Covers. *ICDAR*.

[103] Sochman, J. and J. Matas (2005). WaldBoost ? Learning for Time Constrained Sequential Detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*.

[104] Srivastav, A. and J. Kumar (2008). Text detection in scene images using stroke width and nearest-neighbor constraints. In *TENCON 2008 - 2008 IEEE Region 10 Conference (TENCON)*.

[105] Subramanian, K., P. Natarajan, M. Decerbo, and D. A. Castañón (2007). Character-Stroke Detection for Text-Localization and Extraction. *ICDAR*.

[106] Sun, Q., Y. Lu, and S. Sun (2010). A Visual Attention Based Approach to Text Extraction. *ICPR*.

[107] Sung, K. K. and T. A. Poggio (1998). Example-Based Learning for View-Based Human Face Detection. *IEEE Trans. Pattern Anal. Mach. Intell.*.

[108] Takacs, G., V. Chandrasekhar, N. Gelfand, Y. Xiong, W.-C. Chen, T. Bismpigiannis, R. Grzeszczuk, K. Pulli, and B. Girod (2008). Outdoors augmented reality on mobile phone using loxel-based visual feature organization. *Multimedia Information Retrieval*.

[109] Torresen, J., J. W. Bakke, and L. Sekanina (2004). Efficient recognition of speed limit signs. In *The 7th International IEEE Conference on Intelligent Transportation Systems*.

[110] Tsai, S. S., D. M. Chen, V. Chandrasekhar, G. Takacs, N.-M. Cheung, R. Vedantham, R. Grzeszczuk, and B. Girod (2010). Mobile product recognition. *ACM Multimedia*.

[111] Tu, Z., X. Chen, A. L. Yuille, and S. C. Zhu (2003). Image Parsing - Unifying Segmentation, Detection, and Recognition. *ICCV*.

[112] Vapnik, V. (1999). *The Nature of Statistical Learning Theory*. Springer Science & Business Media.

[113] Viola, P. A., M. J. Jones, and D. Snow (2003). Detecting Pedestrians Using Patterns of Motion and Appearance. *ICCV*.

[114] Wang, K., B. Babenko, and S. J. Belongie (2011). End-to-end scene text recognition. *ICCV*.

[115] Wang, X., T. X. Han, and S. Yan (2009). An HOG-LBP human detector with partial occlusion handling. *ICCV*.

[116] Wang, X., L. Huang, and C. Liu (2009). A New Block Partitioned Text Feature for Text Verification. In *2009 10th International Conference on Document Analysis and Recognition*.

[117] Ye, Q., Q. Huang, W. G. 0001, and D. Zhao (2005). Fast and robust text detection in images and video frames. *Image Vision Comput.*.

[118] Zagoruyko, S. and N. Komodakis (2015). Learning to compare image patches via convolutional neural networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[119] Zhang, J. and R. Kasturi (2008). Extraction of Text Objects in Video Documents: Recent Progress. In *2008 The Eighth IAPR International Workshop on Document Analysis Systems (DAS)*.

[120] Zhang, J. and R. Kasturi (2010). Text Detection Using Edge Gradient and Graph Spectrum. *ICPR*.

[121] Zhang, J. and R. Kasturi (2011). Character Energy and Link Energy-Based Text Extraction in Scene Images. In *Computer Vision – ACCV 2010*.

[122] Zhu, L., C.-S. Yang, and J.-S. Pan (2016). Detection and Recognition of Speed Limit Sign from Video. *ACIIDS*.

Appendix A

Ethics Clearance

Appendix B

Prominence Ranking Survey Results