Text Detection and Recognition from Natural Images

by
Hanaa Fathi Mahmood

A Doctoral Thesis
Submitted in partial fulfilment of
the requirements for the

award of

Doctor of Philosophy of Loughborough University

February 2020

Abstract

Text detection and recognition from images could have numerous functional applications for document analysis, such as assistance for visually impaired people; recognition of vehicle license plates; evaluation of articles containing tables, street signs, maps, and diagrams; keyword-based image exploration; document retrieval; recognition of parts within industrial automation; content-based extraction; object recognition; address block location; and text-based video indexing. This research exploited the advantages of artificial intelligence (AI) to detect and recognise text from natural images. Machine learning and deep learning were used to accomplish this task.

In this research, we conducted an in-depth literature review on the current detection and recognition methods used by researchers to identify the existing challenges, wherein the differences in text resulting from disparity in alignment, style, size, and orientation combined with low image contrast and a complex background make automatic text extraction a considerably challenging and problematic task. Therefore, the state-of-the-art suggested approaches obtain low detection rates (often less than 80%) and recognition rates (often less than 60%). This has led to the development of new approaches. The aim of the study was to develop a robust text detection and recognition method from natural images with high accuracy and recall, which would be used as the target of the experiments. This method could detect all the text in the scene images, despite certain specific features associated with the text pattern. Furthermore, we aimed to find a solution to the two main problems concerning arbitrarily shaped text (horizontal, multi-oriented, and curved text) detection and recognition in a low-resolution scene and with various scales and of different sizes.

In this research, we propose a methodology to handle the problem of text detection by using novel combination and selection features to deal with the classification algorithms of the text/non-text regions. The text-region candidates were extracted from the grey-scale images by using the MSER technique. A machine learning-based method was then applied to refine and validate the initial detection. The effectiveness of the features based on the aspect ratio, GLCM, LBP, and HOG descriptors was investigated. The text-region classifiers of MLP, SVM, and RF were trained using selections of these features and their combinations. The publicly available datasets ICDAR 2003 and ICDAR 2011 were used to evaluate the proposed method. This method achieved the state-of-the-art performance by using machine learning methodologies on both databases, and the improvements were significant in terms of Precision, Recall, and F-measure. The F-measure for ICDAR 2003 and ICDAR 2011 was 81% and 84%,

respectively. The results showed that the use of a suitable feature combination and selection approach could significantly increase the accuracy of the algorithms.

A new dataset has been proposed to fill the gap of character-level annotation and the availability of text in different orientations and of curved text. The proposed dataset was created particularly for deep learning methods which require a massive completed and varying range of training data. The proposed dataset includes 2,100 images annotated at the character and word levels to obtain 38,500 samples of English characters and 12,500 words. Furthermore, an augmentation tool has been proposed to support the proposed dataset. The missing of object detection augmentation tool encroach to proposed tool which has the ability to update the position of bounding boxes after applying transformations on images. This technique helps to increase the number of samples in the dataset and reduce the time of annotations where no annotation is required.

The final part of the thesis presents a novel approach for text spotting, which is a new framework for an end-to-end character detection and recognition system designed using an improved SSD convolutional neural network, wherein layers are added to the SSD networks and the aspect ratio of the characters is considered because it is different from that of the other objects. Compared with the other methods considered, the proposed method could detect and recognise characters by training the end-to-end model completely. The performance of the proposed method was better on the proposed dataset; it was 90.34. Furthermore, the F-measure of the method's accuracy on ICDAR 2015, ICDAR 2013, and SVT was 84.5, 91.9, and 54.8, respectively. On ICDAR13, the method achieved the second-best accuracy. The proposed method could spot text in arbitrarily shaped (horizontal, oriented, and curved) scene text.

Acknowledgment

The chain of my gratitude begins with the name of Allah Almighty, the Most Gracious, and the Most Merciful whose blessings have always been with me.

Special thanks are offered to my parents, brothers and sisters for their continuous encouragement. It would not be possible to achieve all the success without my family. My parents deserve special mention for their unconditional, inseparable support, prayers and endless love.

Special thanks go as well to my husband Professor Rajab Sharief for his incomparable academical and personal support, courage and prayers in the difficult times

I gratefully appreciate and acknowledge the funding and sponsorship of the Iraqi Ministry of Higher Education & Scientific Research (MOHESR) that made my Ph.D. work possible.

I would like to thank my supervisors, Dr. Baihua Li and Prof. Eran Edirisinghe, for their sustained support and valuable advice.

I shall also like to convey thanks to the Computer Science Department, Loughborough University, I'd like also to thank all of my friends and colleagues for their support and encouragements.

Table of Contents

List of Figures

List of Tables

Abbreviations

BLSTM	Bidirectional Long Short-Term Memory
CNN	Convolutional Neural Network
CRF	Conditional Random Field
CRNN	Convolutional Recurrent Neural Network
CTC	Connectionist temporal classification
FPN	Feature Pyramid Network
GLCM	Grey Level Co-occurrence Matrix
HoG	Histogram of Gradient
ICDAR	International Conference on Document Analysis and Recognition
LBP	Local Binary Pattern
MLP	Multi-Layer Perception
MSER	Maximally Stable Extremal Regions
NMS	Non-Maximum Suppression
OCR	Optical Character Recognition
RBF	Radial Basis Function
ResNet	Residual Neural Network
RF	Random Forests
RNN	Recurrent Neural Network
RoI	Region of Interest
RPN	Region Proposal Network
SSD	Single Shot Multi Box Detector
SVM	Support Vector Machine
VGG	Visual Geometry Group
YOLO	You Only Look Once

Chapter 1
Introduction

1.1 Introduction

Our existence is including a huge amount of information in the form of texts, which is read nearly unconsciously as we move within our environment. It has become merged into our lives so much that we are unable to acknowledge its value (Lam et al., 2014).

"In any state or place, a text comprises more information associated with the place and assists us in comprehending the aim with less difficulty" (Lam et al., 2014).

text was classified by Sir John Goody, an anthropologist, as "the most important technological development in the history of humanity" (Lam et al., 2014). Although people have been writing for almost six millennia, machines have achieved substantial progress in reading during the previous century. In conventional optical character recognition (OCR), a printed manuscript is scanned to an image and converted into some machine comprehensible text format. Even though researchers have achieved considerable progress, as yet machines are unable to equal human reading capabilities (Weinman et al., 2014).

Adding modern sub-fields to document recognition and analysis area led to swift expansion this area. The new inventions that increased the number of challenges to the field include: printing, scanning, photocopying and image capturing technology. The notion of recognizing the entire contents of a document is wonderful idea but it still challenges to attain using a machine. Nonetheless, the fundamental version of a photocopying machine does have this ability to some extent. Recently machines practically carry out this task by generating a reasonable soft copy of the document which can be obtained by current OCR (optical character recognition) techniques (Lienhart and Wernicke, 2002).

There are a large amount of ideas and methods for localization and extraction of textual information from natural images and videos.

In general, these approaches can be divided in to three categories:

1-Text detection: Text detection aims to generate candidate text regions from natural scene images that correspond to texts. Text detection distinguishes the text areas and create candidate bounding boxes from image. Text localization is determination the text location in image and draw bounding boxes around the text. The text recognition phase extracts the text information from such bounding boxes areas. Although bounding boxes specified the accurate location of

text in an image, segmented text from the background still needed. This process includes transform image to a binary image and enhance it before it is fed into an OCR engine. Text extraction is the stage of segmented text region from background. Usually the segmented area has different type of noise and low resolution for this reason it required a number of enhancement operations. Furthermore, extracting the contain from images is considerably difficult, due to image quality and background noise. Thereafter, OCR used to transform extracted text images to plain text see Figure1.1 (Wang et al., 2015)(Zhu, Yao and Bai, 2016).

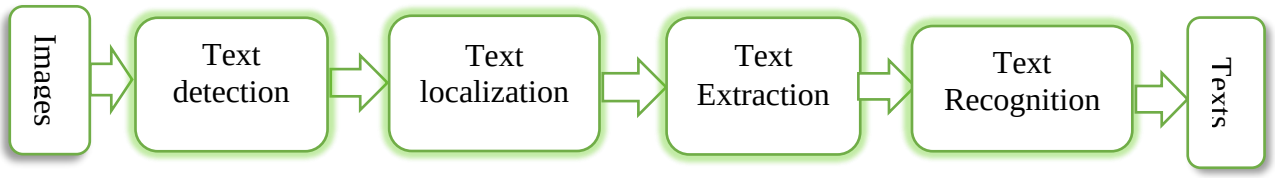


Figure 1.1: Architecture of an overall text detection and extraction system.

Regarding scene text images, an enhancement of more than 50% of text recognition rate may be obtained through the improvement of text detection (Gatos et al., 2005). Thus, text detection is essential for text information retrieval (Wang et al., 2015).

2-Text recognition: Text recognition converts detected text regions into strings. In recent research, word recognition has been central to text recognition because words are well-formulated with statistical models in terms of low-level features and high-level language priors (Ye and Doermann, 2015).

3-End-to-End: When the input source constitutes an image with a complex background, the technique used to identify it will be formed of an end-to-end technique composed of processes for localizing, detecting and recognising all text and transforming that text in the image into strings. Working with a brief vocabulary list, the use of word spotting affords an efficient approachs for achieving end-to-end recognition. As" the whole is greater than the sum of parts" this is clear inspiration for employing word spotting where the given activity involves matching a particular word in a specified lexicon with its corresponding image patches through the utilization of word and character examples. In the case of open lexicons, the region included in the search is sizeable and word spotting approaches are unfeasible. To address this matter the process used must employ more powerful character identification abilities (Wang et al., 2012), substantial language examples (Bissacco et al., 2013) and advanced optimization strategies (Weinman et al., 2014).

- ## Motivation and Overview of Text Detection and Recognition Applications

Text detection and recognition from images could have numerous functional applications for document analysis such as assistance for visually impaired people, recognition of vehicle license plate, evaluation of article comprising tables, street signs, maps, diagrams, keyword based image exploration, document retrieving, recognition of parts within industrial automation, content based extraction, object recognition, address block location as well as text based video indexing, (Wu, Manmatha and Riseman, 1997) (Seeri, Pujari and Hiremath, 2015).

Several applications could benefit from text detection and recognition.

- **Document processing:** the aim is to convert different types of text available in scanned documents, web images, video images, and digital cameras into a different electronic version such as a word processing document.
- **Web indexing:** most web images have embedded text that increases the semantic content value of the image. This text can be used to index the images, thus making web page indexing more convenient.
- **Retrieval and indexing of video images:** This is useful when arranging movies based on content. The rapidly increasing demand for streaming video services such as YouTube requires effective structures to index and extract textual context.
- **Automated reading:** This may be employed to assist blind or visually impaired people to read text included in digital images. A considerable amount of applications are available for automated narration of documents for individuals such as car drivers (Jung, Kim and Jain, 2004).
- **Screenshot OCR:** Modern word processing software provides the alternative to build screenshots from sections of a document. This alternative is often employed as a swift and simple method to publish certain sections of a document.
- **Object/product identification:** Acquires additional information regarding a product or item.

Furthermore, A recently invented application is the mobile banking facility offered by the banks, which enables the customers to perform transactions even on sending the image of a cheque to the server. Another similar application is tourist guide, which enables the tourists to comprehend display boards even though they are not conversant with the language of the area, and image text conversion structures assist blind people as well as tourists. Each such application is dependent on a TIE (Textual Information Extraction) structure which can

competently detect, localize and retrieve the text information available in the natural images (Jung, et al., 2004).

Moreover, Scene text detection and recognition techniques can utilize in detecting text-based landmarks, vehicle license detection/identification, and object recognition as opposed to overall extraction and indexing. It is a challenge to detect, locate and recognize scene text as there could be unlimited possible fonts, sizes, colours and shapes, resolution, intricate backgrounds, uneven lighting, or blurring resulting from differing lighting, intricate movement and conversion, unfamiliar format, shadowing as well as differences in font size, style, alignment and direction. Further, texts may be in various scripts (Seeri, Pujari and Hiremath, 2015). Text extract from images or videos can be exploited to annotate and index those materials. For example, extracting the text from player's uniform in the sequences of video game can be used to annotated and indexed videos (Wu, Manmatha and Riseman, 1997).

1.2 Major Challenges of Text Detection and Recognition

Text detection is the challenge of finding text within an image. Provided any image, a text detector must return all the bounding boxes of all text within the image. No suppositions are made regarding the size, direction or position of the text.

Researchers have suggested several techniques of discovering texts in natural images and videos because of its possible uses in numerous applications. It is a difficult task as a result of the rapid rise in the digitalization of all the materials. Within a natural scene, text extraction generates practical and valuable information which may be recognised by both, humans and computers. Text detection and recognition is a considerably active and challenging task in the field of computer vision.

In natural scene images, text detection and recognition are challenging for the same reasons as text identification which include factors such as textured backgrounds, difference in text resulting from lighting conditions, deformation due perspective view, and images of reduced quality.

Therefore, within natural scenes, detection and recognition of texts is a considerably challenging task. Those challenging come from the properties of text in the image. The main challenges for scene text detection as well as recognition may be roughly classified into the following types (Yao, Zhang, et al., 2013) (Yao, Bai, et al., 2013):

- **Diversity of scene text**: Text in document images normally appears in regular font, single colour, and uniform layout. In contrast, texts from natural scenes may have

completely different fonts, colours, orientations and scales even within one scene(Liu et al., 2011). Characters from different fonts may include large within-class differences and may create numerous sequence sub-spaces, making it a challenge to carry out precise recognition if the character class number is big (Liu et al., 2011).

- **Scene complexity**: In natural environments, numerous man-made objects, such as buildings, symbols and paintings appear within natural scene images and videos which may have similar structures and appearances to text. Thus, the backgrounds may be quite complex. Text itself is generally laid out to enable readability. However, the problem with scene complexity is that the encompassing scene makes it a challenge to distinguish text from non-text.
- **Aspect ratios:** Text has varying aspect ratios within a natural scene. For instance, traffic signs could be brief, but other text, such as video captions, could be considerably longer. Detecting this types of texts require a procedure with a high computational complexity since the location, scale and length needs to be considered within the search process (Yao, Zhang, et al., 2013) .
- **Interference factors**: There are different interference factors, such as the following:

  Uneven lighting: When taking pictures in a natural setting, uneven lighting is a common problem as a result of varying reaction of sensory devices to the light source. Uneven lighting usually leads to erroneous colour information and degradation of visual elements, and as a result leads to false detection, segmentation and identification outcomes.

  Blurring and degradation: Blurring of text and defocusing of images happen even in optimum working circumstances and focus-free cameras. Additionally, other factor degrades the standard of the text such as image/video compression/decompression especially in the case of graphical video text. The overall impact of degradation, blurring and defocusing is that they reduce the precision of characters and lead to “touching” of characters, which makes segmentation a challenge (Liang, Doermann and Li, 2005).

  Distortion: When the camera optical axis is not perpendicular to the text plane, perspective distortion takes place; This lead to loss of text boundaries, rectangular shapes and deformed characters thereby reducing the functioning of identification models trained using undistorted pattern (Zhu, Yao and Bai, 2016) (Liang, Doermann and Li, 2005).

Noise : Reduced resolution (Zhu, Yao and Bai, 2016).

- **Multilingual environments**: Even though the majority of the Latin languages have tens of characters, other languages include thousands of character categories such as Chinese, Japanese and Korean (CJK). Arabic characters change shape according to context because it has connected characters. Hindi integrates alphabetic letters into thousands of shapes which signify syllables. Within multilingual settings, OCR in scanned documents is still a research challenge (Smith, 2011), although text identification within complex imagery is additionally challenging(Smith, Antonova and Lee, 2009).

Regarding scene text images, an enhancement of more than 50% of text recognition rate may be obtained through the improvement of text detection(Gatos et al., 2005). Thus, text detection is essential for text information retrieval (Wang et al., 2015).

1.3 The Aim and Objectives

Detecting and recognising text involves a very delicate procedure, which detects a number of varying fonts, contrasts, font sizes, alignments, and colours. Each of these elements makes it challenging to establish a single solution. An additional challenge is the fact that humans are regarded as a benchmark for determining whether the algorithm functions in a similar manner to our eyes and brain, as at times there are challenges related to the detection and recognition of text in videos and images.

The main aim of the research presented in this thesis was to develop a robust text detection and recognition system from natural images with high accuracy and recall, which would be used as the target of the experiments.

The above aim was met by following the objectives listed below:

- Analyse the literature review on text detection and recognition from natural images, which can explain the methods used and identify the features of each method. Review and compare traditional and current deep learning-based algorithms for detecting and recognising text in natural images. Furthermore, Demonstrate the evaluation protocols and the state-of-art existing benchmarks.
- Identify the most significant challenges and applications.
- Develop algorithms to identify the location and recognise text in natural images
- Generate a new dataset, which can fill the gap of character annotation, multi-orientation, and curved text. Furthermore, develop image augmentation

methods and tools on the bounding-box level to provide more samples for training, thus reducing the manual annotations of a large number of new images.

- Implement the proposed methodologies and algorithms and carry out an experiment on the proposed dataset. Evaluate the proposed methods and compare the results with the previous methods.

1.4 Key Contributions

In this study I have manage to build successful detection and recognition frameworks which have been proposed in the field of scene text detection and recognition. A number of methods in various areas of the detection and recognition framework are proposed, as successfully tackled the challenges of text detection and recognition.

The research conducted within the context of this thesis has led to the following original contributions.

1- A Robust and an efficient method has been proposed and implemented to detect text in scene images by using multi-feature and multi-classification techniques. The MSER technique was used to detect a blob area in a scene image. The MSER-detected regions contained both text and non-text regions. Then, a combination and selection of HOG, LPB, GLCM, and aspect ratio features were used as the descriptors to represent the text candidate regions. SVM, MLP, and RF were used as classifiers to filter the text regions. All the possible combinations between the used features were tested to achieve the best detection accuracy. The selection and the combination of features are the two main contributions were proposed in this study. The novel technique applied in this study uses a small set of heterogeneous features which are especially combined to build a large set of features. By comparison with the previous methods those used single feature descriptor, the proposed techniques applied in my study has improved the detection accuracy and helped to remove the non-text regions. The results showed that the use of a suitable feature selection and combination approach could significantly increase the accuracy of the methods and reduce the implementation time. The proposed methods applied have achieved the state-of-the-art performance on two datasets, namely ICDAR 2003 and ICDAR 2011 in terms of Precision, Recall, and *F*-measure. The *F*-measure for ICDAR 2003 and ICDAR 2011 was 81% and 84% respectively

2- Create a new dataset of English text in natural images. This is a challenging dataset with a good diversity, going considerably beyond the previous datasets. It has been

proposed to fill the gap of character-level annotation, the number of samples in the exists datasets and the availability of text in different orientations. Most of the available datasets suffer from the lack of samples for training, especially those used for deep learning approaches. Moreover, the existed dataset lack diversity and variation texts, most of datasets are specialized in one orientation of texts such as horizontal, multi orientation or curve. They are annotated in the line or word level; character level annotation is not available especially for detection task. The proposed dataset was created particularly for deep learning methods which require a massive completed and various range of training data. The dataset contributed in this study includes 38,500 images of English characters and 12,500 words in more than 2100 images. It is contained text in an arbitrary shape (combination of horizontal, multi-oriented, irregular and curved text). The proposed dataset annotated by myself in character level and word level. I believe this is the first dataset that produces digit annotation along with character annotation, while most of the existing datasets ignore digit annotation. Furthermore, my other contribution is the proposed of augmentation tool which is created to support the proposed dataset due to the missing of the augmentation tool for object detection tasks. The position of the bounding boxes needs to be updated for object detection augmentation. Therefore, this study provided an augmentation tool along with the proposed dataset for bounding boxes augmentation without the requirement for annotating new images, the position of the bounding boxes and the class can be obtained automatically from the original image. This technique helped to increase the number of samples in the dataset and reduce the annotation time, no annotation was required.

3- A new robust method for text detection and recognition to overcome the limitation of the previous methods, by detecting and recognizing each character individually which is helping to avoid the accumulation of intermediate error and accelerates the processing speed. Furthermore, the proposed method able to detect and recognize characters simultaneously, it is a unified deep neural network which is a single forward pass. The proposed framework is a trainable end-to-end character detection and recognition system designed using an improved SSD convolutional neural network. The traditional SSD network is unskilled at dealing with small objects that due to having same kernel size for all layers. Furthermore, the aspect ratio used to detect objects dose not suit text objects. The structure of the network has been improved by adopting better feature extractor backbone. Where layers with different kernel sizes are

added to the SSD networks in order to replace the last layers of the original SSD. The aspect ratio of the characters is considered because it is different from that of the other objects. Furthermore, the proposed network has the ability to spot digits alongside with characters. To my best knowledge, this is the first method integrate texts and digits detection and recognition. While other existing methods focuses on either texts or digits although both are available in scene images together. Compared with other methods considered, the proposed method can detect and recognize characters by training the end-to-end model completely. It has also the ability to spot specific texts in arbitrarily shaped (horizontal, oriented, and curved) scene text with high accuracy. The proposed method applied on the proposed dataset achieved great performance with an accuracy of 90.34%. Furthermore, the F-measure of the method's accuracy on ICDAR 2015, ICDAR 2013, and SVT was 84.5%, 91.9%, and 54.8%, respectively. On ICDAR13, the method achieved were the second-best accuracy among the several widely used models.

The above original contributions have resulted in the following paper

- MAHMOOD, H.F., LI, B. and EDIRISINGHE, E.A., 2017. Text localization in natural images through effective re identification of the MSER. IML '17 Proceedings of the 1st International Conference on Internet of Things and Machine Learning, Liverpool, United Kingdom, October 17th-18th 2017, article no.42

1.5 The Structure of the Thesis

This thesis is organized into seven chapters and the summaries of each chapter are as follows:

Chapter 1: presents an introduction to the research field. A brief description of the study motivation states the importance of text detection and recognition, and the procedures to extract text information. It shows the types of texts presented in scene images and it also illustrates properties of text in images followed by the application of text. This chapter also explains the major challenges in text detection and reviews the related works on scene text detection and recognition It also explains the motives to undertake research in this area.

Chapter 2: is a review of the background and literature review. It explains the methods used for text detection and recognition and illustrates the advantage and disadvantage of each method. This chapter divides the methods in two parts which are methods used machine learning and methods used deep learning. This chapter refers to the stat of art of previous methods.

Chapter 3: includes an explanation of the basic algorithm and its implementations. This chapter describes the theories that used to extract features and built descriptors that uses in chapter four. The theory of used descriptor such as HOG, LBP, GLCM and classifiers such as SVM, RF are included in this chapter.

Chapter 4: explains the steps of implementation of the proposed method. The flowchart of the training and testing phases that used to classify MSER region and the results.

Chapter 5: explains the details of the proposed dataset and the augmentation tool. This chapter includes the properties of the existed dataset and the evaluation protocols that used to calculate the performance of the methods. The chapter reviews the contains of the proposed dataset, number of images, types of images, type of text in the images. Furthermore, the annotation methods and annotation level are explained. Furthermore, chapter five introduces the augmentation tool that has been used to increase the samples of proposed dataset.

Chapter 6: explains the proposed framework of the end-to-end character detection and recognition system designed with improved SSD convolutional neural network. The structure of the proposed network has been explained in detail. The experiments and results have been discussed in detail.

Chapter 7: concludes with a view to further improvements of the proposed algorithms and suggestions for future research.

Chapter 2

literature Review

2.1 Introduction

Optical Character Recognition (OCR) has been viewed as solved issue by many researchers. But, text detection and recognition in imagery suffer from many of constraints as computer vision and pattern recognition problems such as lower quality or degraded data. Therefor state-of-the-art of suggested approaches obtain the low detection rates (often less than 80 %)(Yin, Huang and Hao, 2013) and recognition rates (often less than 60 %) .That encroach to propped new approaches. Where, the recognition rates of typical OCR on scanned documents achieved recognition rates higher than 99% (Weinman, Learned-miller and Hanson, 2009)(Neumann and Matas 2013)(Ye and Doermann, 2015).

The field of text detection and recognition have different problem which requires application of advanced computer vision and pattern recognition techniques. Which are Complex backgrounds, low resolution, uneven illumination, different fonts, and variations of text layout.

The demand for information extraction has been increased due to the affordable of cameras and the rapid growth in digital technologies and devices equipped with megapixel cameras. As well as the invention of the modern touch screen technique in digital gadgets such as PDA, mobile and others result in numerous new study challenges. Such as a camera captured image which is inclined to encompass more of a scene than text. Where textual information may or may not be present in many scenes (Kim, 1996).

- **Text in Images**

Within the field of computer vision and pattern recognition, recent studies have shown considerable interest in content extraction from videos and images (Seeri, Pujari and Hiremath, 2015). Techniques in content-based image indexing outline the procedure of affixing labels to images subject to their content. This content could be in the form of objects, colour, texture, and shape. Associations linking these types of content are also included. Image content may be classified into two main categories: perceptual and semantic (Kim, 1996) (Zhong and Jain, 2000).

Perceptual content encompasses attributes such as texture, colour, shape, intensity, and alignment of text in addition to temporal modifications. A number of techniques have been published on the use of comparatively low–level perceptual content for video and image indexing (Kim, 1996) (Zhong and Jain, 2000).

On the other hand, semantic content refers to events and objects, as well as their associations. Semantic content delivered by an image could be practical for content-based image retrieval, in addition to classification and indexing purposes (Zhong and Jain, 2000) (Seeri, Pujari and Hiremath, 2015). Recent research has focused on semantic image content to detect face, human, and vehicle activity (Lienhart and Wernicke, 2002)(Strouthopoulos, Papamarkos and Atsalakis, 2002).

Different font styles, sizes, alignment, and colours could be embedded within an image in the form of text data against a complex background. The challenge of retrieving the candidate text area, which is available within a natural scene image, becomes a difficult one. Rapid advancement of digital multimedia technology has led to the digitisation of all classes of information resources, which are usually accessible electronically (Seeri, Pujari and Hiremath, 2015).

- ## Types of Text in Scene Images

Two classes of text may be found in scene images:

Scene text: available in natural scenes such as names of shops, street signs, street names (within a city scene comprising numerous shops and streets), and text on a surface such as a T-shirt and flags. These texts are a challenge to detect and identify since the text may be tilted, skewed, rotated, partially hidden, partially shadowed, or laid on various surfaces (Ye and Doermann, 2015).

Graphic text: artificial or caption text that is usually added over the scene in the post-processing stage. This includes embedded captions in TV programs, newscasts, subtitles, and video annotations. Digital images such as news videos, video commercials, and sports videos are also included. This class of text has the additional advantage of being mostly straight since it is added in rectangular frames to video images (Zhang et al., 2013).

Artificial text: produced and laid over the scene in the post-processing stage is often an important carrier of information and is therefore suitable for information indexing and retrieval (Lienhart and Wernicke, 2002).

Graphic text, which usually appears in clusters and lies horizontally, does not have perspective deformations. This type of text is usually monochromic but may have shadows for effective visualisation (Zhang et al., 2013).

▪ Properties of Text in Images

Text is often displayed with varying degrees of size, front, alignment, style, orientation, contrast colour, background, and texture (Jung, Kim and Jain, 2004). The following outlines a list of attributes that have been employed in recently published algorithms (Jung, Kim and Jain, 2004)(Zhang et al., 2013)(Zhu, 2015) (Zhu, Yao and Bai, 2016). Text in images may display numerous distinctions with regard to the subsequent features:

- **Geometry:**
  - **Size:** text size can differ considerably since it is application-dependent.
  - **Alignment:** one of the features of captioning text characters is to produce a horizontal appearance in clusters, although some special effects may lead to characters being displayed as non-planar texts. This is not applicable to scene text, which could comprise different perspective distortions. Scene text can have geometric distortions and may be aligned in any direction.
- **Edge:** The designers of a caption and scene texts aim to display easily readable text. This aim is achieved by displaying texts in strong edges at the boundaries between the text and the background.
- **Colour:** this feature encouraged researchers to use a connected component-based approach for text detection whereby the characters within a text line are inclined to have similar or the exact same colours. Most of current research has focused on determining 'text strings of a single colour (monochrome)'. Nevertheless, video images and alternative complex colour documents may comprise 'text strings with several colours (polychrome)' for efficient visualisation, i.e., various colours inside one word.
- **Compression:** in order to reduce the size of images and the time taken to transfer and process images, researchers have compressed images using different methods.
- **Motion:** in a video, the same characters are often present in sequential frames with or without motion. This feature is utilised in text tracking and improvement. Text within video usually moves in a uniform manner, either vertically or horizontally. However,

scene text may comprise random movement due to object or camera motion (Jung, Kim and Jain, 2004).

2.2 Methodologies of text Detection and Recognition:

Comprehensive processes for text detection and recognition are broadly categorized into two types; integrated techniques and stepwise techniques:

- **Integrated techniques**: This technique seeks to recognize words using detection and recognition processes together with character classification systems that distribute data between them and/or employ combined development approaches as illustrated in Figure 2.1.
- **Stepwise techniques:** consisting of detection and recognition modules that are disconnected, and they detect, segment and recognize text areas through the use of a feed-forward pipeline is illustrated in Figure 2.2.

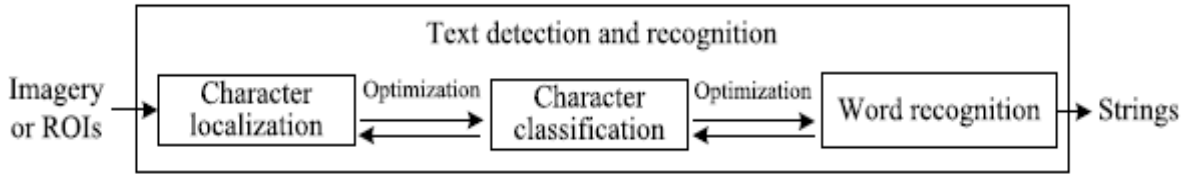


Figure 2.1: Integrated methodology (Ye and Doermann, 2015)

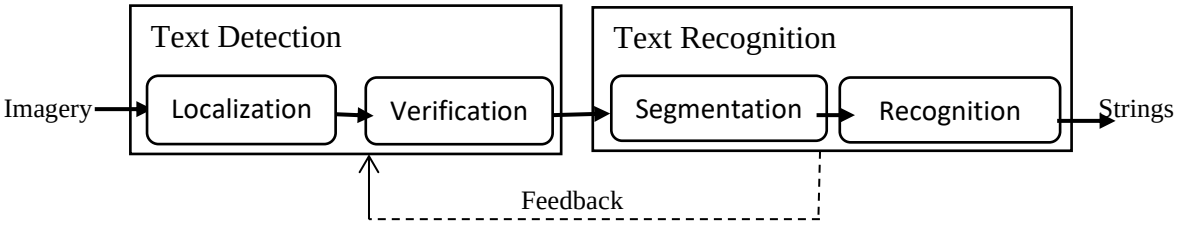


Figure 2.2: Stepwise methodology (Ye and Doermann, 2015)

Integrated techniques utilize recognition as the primary means. Moreover, several of the strategies of the methods employ a pre-processing stage that helps localize regions of interest. This differs from stepwise techniques that seek to reduce the false detection rate by taking advantage of a text recognition feedback process (Ye and Doermann, 2015).

The most crucial quality of stepwise techniques is their use of coarse localization as this allows the filtration of the majority of the background details. The benefits realized here are two-fold; they increase the efficacy of the computational system while reducing its associated expenses. Disadvantages, however, do exist and include the complicated structure of the technique as this technique differs from others that integrated all the stages together. Moreover, it is not easy to ensure all factors for all stages are optimized and so this technique is abstracted by the build-up of errors.

Integrated techniques bypass the issues brought by segmentation as they seek to recognize specific words using examples of character and languages models. These techniques, however, are disadvantaged by the expensive computation cost involved for multi-class character classification in cases where extensive character classes and extensive candidate windows are employed. Moreover, the greater the word class number the worse the detection and recognition ability of the process. Finally, these techniques are also restricted by their small size of the lexicon list.

Technique qualities

- Stepwise techniques

  Distinct detection and recognition modules.

  Process split into localization, classification, segmentation and recognition phases.

  Appropriate for the detection of a sizeable number of words.

  Low computational expenses though, but the higher complexity of the process due to additional stages.

- Integrated techniques

  Combined detection and recognition modules.

  Segmentation step is eliminated and replaced with a stage for word recognition.

  Appropriate for the recognition of certain words from images due to small lexicon

  The larger the lexicon list the harder it is to identify words. (Ye and Doermann, 2015)

2.3 Related works

This section will discuss and assess both the standard and developed algorithms used in the detection and recognition of text. It will also discuss and compare the methods of deep learning techniques.

2.3.1 Scene Text Detection and Recognition Using Machin learning

1- Text Detection

Text detection comprises a very dynamic region of research. For scene text images, an enhancement of more than 50% in rate of text identification may be attained through the use of text detection (Gatos et al., 2005). Thus, text detection is essential for text information retrieval(Wang et al., 2015). Regarding text localization, CCA (connected component analysis) and sliding window classification comprise two broadly employed techniques, and the typically employed features encompass texture, edges, strokes and color (Ye and Doermann, 2015).

Sliding window-based methods:

Sliding window-based methods, alternatively referred to as region-based techniques have proved to be considerably effective for applications such as pedestrian and face detection. Character detection comprises exceptional challenges, though comparable to applications mentioned above. Addressing a great quantity of classes, such as 62 classes for English characters, which is the primary challenge. The next problem is the confusion of the inter-characters and intra-characters as outlined by Figure 2.3. When a window comprises parts of two characters close to one another, it could lead to an appearance similar to that of an alternative character. The window in Figure 2.4 (a) includes sections of characters ‘o’ which may be consider as ‘x’. Additionally, the appearance of a section of a character may resemble another character. A section of the character ‘B’ may be consider as ‘E’ in Figure 2.3(b) (Koo and Kim, 2013).

a b

Figure 2.3: The challenges in detect multi-class character detection(Koo and Kim, 2013). .

(a) Inter-character confusion: A window covering parts of the two o’s is incorrectly detected as x.

(b) Intra-character confusion: A window covering a segment of the character B is recognized as E.

Essentially, sliding window consider as a brute-force technique, which seeks potential text within the windows by means of a sliding window and subsequently recognizes text by means of machine learning methods. As the image requires processing through multiple scales, these techniques are slow.

Thus, for region-based methods the focus of research has been on developing an effective binary classification (text against non-text) for a minor image patch. Alternatively stated, the focus has been on the problem of the Determine if a provided patch comprises part of a text area issue:(Koo and Kim, 2013) (Yin, Huang and Hao, 2013).

Researchers tackled this challenge through implementation of cascade structures for efficient classification. Simple features such as vertical and horizontal derivatives were employed during the initial phases of the cascade for their techniques, and complex features were incrementally used (Koo and Kim, 2013). This still remains a problem although proficient text detection is facilitated by this structure. Even for a human, it is difficult for the class of a minor image patch to be determined when the text properties are unknown including color, skew and scale. Multi-scale scheme using different window sizes can reduce the scale issues; nonetheless this causes the overlapping of boxes in varying scales. This region based technique has been observed to be efficient from experimental outcomes using ICDAR 2005 database, nonetheless returning a worse performance when compared with CC-based approaches (Koo and Kim, 2013)(Pan, Hou and Liu, 2011).

This method is an easy and adaptive training-detection architecture, however, when there is use of complex classification techniques and a great amount of windows requiring classification, it is usually computationally expensive (Ye and Doermann, 2015)

Chen and Yuille used three sets of features over multi-scale image windows and AdaBoost classifier to detect text in cascading method. First set include 40 features which are, 4 intensity mean features, 12 intensity standard deviation features, 24 derivative features, second set consist of 24 features of histograms features, The rest are 25 features based on edge linking. Segmentation is subsequently acquired through employment of a variation of Niblack's adaptive binarization algorithm. As this technique requires manual segmentation for numerous sub-windows for training objectives it is computationally expensive, and its localization performance is not apparent due to the lack of standard evaluation protocol used.

Pan, Hou and Liu, incorporated a combination of multi-scale Local Binary Pattern and HOG (Histogram of Gradient) feature. Then a connected component analysis (CCA) method has been employed to filter out non-text CCs, based on Markov Random Fields (MRF) model. ICDAR 2003 datasets has been used for training and ICDAR 2005 datasets for testing. The performance of Pan, Hou and Liu, method achieved 0.68% for Recall and 0.67% for Precision. However, this technique showed a higher computational complexity due to use of multi scale windows (Pan, Hou and Liu, 2008).

The technique was recently enhanced by (Lee et al., 2011) through the inclusion of additionally discriminative and more computationally costly features A whole image was sequentially searched by employing multi-scale windows to adapt different scene text sizes, and six effective features were retrieved for every window(X-Y derivatives, edge interval and analysis of connected components, local energy of Gabor filter, statistical texture measure of image histogram, measurement of variance of wavelet coefficient) and input into a modest AdaBoost for classifying the related areas as non-text or text. The method achieved 0.70% for f-measure on ICDAR 2005. Text localization performance was enhanced slightly, although the technique was considerably slowed down.

Component based methods:

A bottom-up technique is employed by connected component-based techniques through the grouping of small components to progressively greater components until the identification of all regions within the image is achieved. In the subsequent phases, a geometrical analysis is usually necessary for the identification and grouping to localize text regions of text components. Candidate text components are immediately segmented using color clustering or edge detection in CC-based techniques. Classifiers or heuristic rules are subsequently employed to prune the non-text components. CC-based techniques may lead to a reduced computation expense due to the comparatively minor amount of segmented candidate components, and recognition may be carried out immediately using the located text components (Yi and Tian, 2011) (Zhu, Yao and Bai, 2016).

Nonetheless, text components cannot be segmented precisely by CC-based techniques in the absence of previous knowledge of text scale and position. Additionally, it is a challenge to design dependable and fast connected component analysers due to the numerous non-text components which are easy to be confused with text when subjected to individual analysis. In general terms, due to the comparatively small number of components that require processing

these techniques are considerably more efficient. These techniques are additionally insensitive to font variation, scale change or rotation. Within the scene text detection field, component based techniques have become conventional of recent years (Yi and Tian, 2011)(Huang et al., 2013).

CC-based techniques comprise CC extraction and localization of text regions through the processing of only CC-level information. Their concentration is thus based upon the following challenges:

Problem (A): extraction of text-like CCs

Problem (B): filtering out of non-text-like CCs

Problem (C): inference of text blocks from CCs

The development of numerous techniques of CC extraction within the literature was for Problem (A). Several techniques assumed, for instance, that strong discontinuities would be showed by the text component boundaries and CCs were extracted from edge maps. The observation that text is written using the same color motivated other researchers, who employed methods of color segmentation or reduction. Some researchers alternatively established independent methods of CC extraction from the beginning. For example, (Epshtein, Ofek and Wexler, 2010) utilized text curvilinearity, in addition to local binarization which was applied in (Pan, Hou and Liu, 2011) through the employment of estimated scales. Non-text CCs are filtered out using CC-based techniques following the CC extraction. Ultimately, (Pan, Hou and Liu, 2011)(Chen et al., 2011) utilized features like “the difference of the stroke width within every CC”, “the quantity of holes within a CC”, and “aspect ratio”. CRFs (conditional random fields) were applied in (Pan, Hou and Liu, 2011) so as to extract binary (relational) features in addition to unary features. For the filtering of non-text components, a neural network was employed.

From the remaining CCs, CC-based techniques ultimately infer text blocks. Text line grouping, text line formation or text line aggregation are all terms used to refer to this stage. Several techniques were proposed based on the above. For instance, several techniques have employed color difference and the ratio of height between two letters (Epshtein, Ofek and Wexler, 2010)(Chen et al., 2011).

CC-based techniques often suffer from high computational complexity even though they have shown better performance than region-based ones. This is a result of their performances

being subject to the CCs quality, and their application of refined CC filtering and extraction techniques (Epshtein, Ofek and Wexler, 2010)(Neumann and Matas, 2011).

Wang & Belongie (Wang and Belongie, 2012) suggests a two-stage technique based on multi-layer segmentation and high order CRF (conditional random field) as shown in Figure 2.5 . The technique employs multi-layer segmentation to divide text from its setting using the mean shift algorithm by which the input image is decomposed into nine levels subject to their gradients, colors and contrasts. The CCs (connected components) within these varying levels are acquired as candidate text. High order CRF based evaluation is employed to validate the candidate text CCs. Features from three various levels, comprising CC strings, CC pairs and separate CCs are included using a higher order CRF model to differentiate non-text and text. For easy evaluation, the remaining CCs are subsequently grouped into words. The method is observed to achieve a good performance for natural scene text detection when employing the ICDAR and street view datasets.

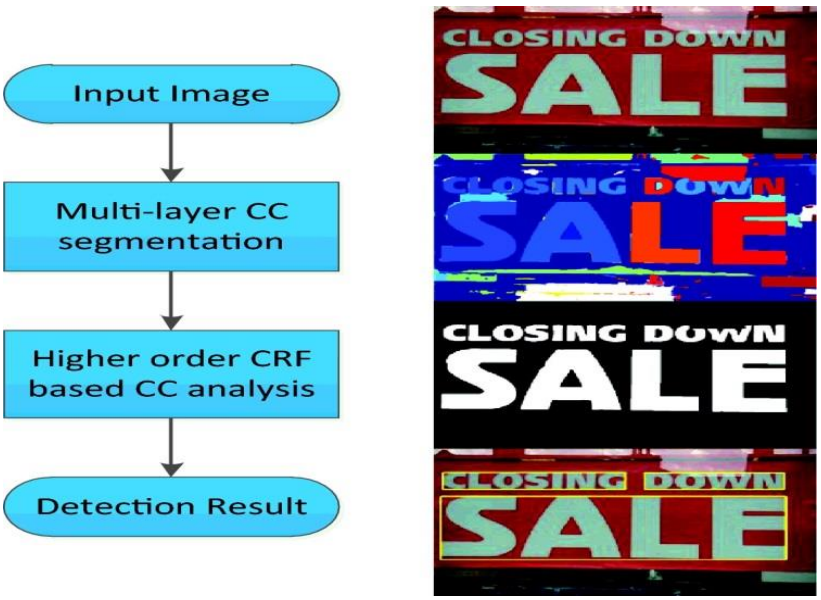


Figure 2.4: The diagram of Wang, X. et al., 2015 method. Applying multi-layer segmentation create CCs which are labelled by different colors and Yellow used to bounding the detected words.

Features

Texture, edge and color features were employed typically for text localization, and features of character appearance, region, point and stroke have been investigated recently (Ye and Doermann, 2015)(Liang, Doermann and Li, 2005).

Color-based methods: Normally, the text produced by designers of scene or caption text is in a recognizable brightness and color and is prominent from the background. Color features may be employed to detect text and text may be split from background with emphasis on varying color characteristics that they possess subject to this supposition (Liang, Doermann and Li, 2005).

Karatzas, D. and Antonacopoulos, A., 2004 (Karatzas and Antonacopoulos, 2004) applied split-and-merge strategy, based on the HLS (Hue-Lightness-Saturation) representation of color as a primary estimation of the variations between lightness and chromaticity. The selection of the HLS color space is due to the factors which facilitate chromatic color separation by humans, mainly comprising the wavelength division amid colors (articulated by Hue variations), the purity of the concerned colors (articulated by Saturation) and the apparent luminance of the concerned colors articulated by Lightness). In addition, the manner in which merging take place within the component aggregation phase is directed by the HLS image data which is employed in connection with the accessible information on Lightness, Color Purity and Wavelength discrimination.

A bottom-up merging procedure is carried out following the conclusion of the splitting procedure. Connected components are initially identified within every one of the bottoms (leaf) layers. character-like components are subsequently retrieved as making up text lines in several directions and alongside curves. 88% recall and 53.4% accuracy are attained by this technique

Gradient-feature-based methods: It is supposed that a strong edge is displayed by text against background, and thus pixels comprising high gradient values are viewed as suitable text region candidates for the family of text localization techniques which are gradient-feature-based. The non-text regions are filtered out using heuristics following the recognition and combination of the edge of the text boundary. For the edge detection, an edge filter is normally employed and for the merging phase a morphological operator is employed. Usually, the techniques start with either edge detection or alternative gradient-based feature computation (Ye and Doermann, 2015).

An edge-based technique for text detection within images comprising text in the horizontal orientation was suggested by Shivakumara, P., Huang, W. & Tan, C.L.,(2008). Sixteen equivalent non-overlapping blocks were segmented from a frame of size 256×256. Heuristic rules were derived from mean and median filter as well as edge analysis to identify the candidate text block for segmenting. These different operations result in $S_{AF}(x,y)$ and $C_{Diff}(x,y)$

where S_{AF} comprises the Sobel edge block for the arithmetic filter result and C_{Diff} is the canny edge block for the variation block ($_{Diff}$) which is subtraction of the arithmetic filter from the median filter of the block. Let NS_{AF} and NC_{Diff} be the quantity of edges in SAF (x,y) and $C_{Diff}(x,y)$, correspondingly. Thus, the subsequent rule:

$$R_1 = \begin{cases} \text{Test Block}, & \text{If } NS_{AF} > NC_{Diff} \\ \text{NON Text Block}, & \text{Otherwise} \end{cases} \quad (2.1)$$

where the entire text block was acquired. Ultimately, the true text regions are detected based on the horizontal and vertical bar feature.

For primary text detection Shivakumara et.al (2009) employed edge analysis and filters. Due to the number of false positives resulting from employment of Canny edge profile for detection of text blocks, straightness and cursiveness edge features being employed for elimination of false positives. This method was contrasted with previous techniques which employed uniform color for text location as well as methods based on the gradient for detection of text, as well as an alternative technique based on Sobel edge features. Compared to the other techniques, the number of blocks detected by this technique is greater as since the technique detects text comprising poor contrast and small font, which the other techniques are unable to do. In addition, detection of text blocks without missing characters is carried out by this technique, and the rate of misdetection is therefore lower compared to other available techniques. They created an individual dataset the results showed that the rate of detection was 89.13% (Shivakumara, Huang and Tan, 2008).

An automatic translation structure was provided by Park, J. et al., (Park et al., 2010) for images on Korean signboards, where a shop name is often signified by the text. They used the fact that the edge component is a reliable feature for identification of text detection due to it being less sensitive to light differences. To obtain the edges of the input image, the canny-edge detector is applied on the gray-scale image. The horizontal profiles of the edges are calculated for detection of the candidates of the area comprising the text. Through vertical scanning, to the lower or upper ends of image, beginning with the image's center line, detection of valleys within the histogram occurs, which results in the identification of the candidate regions using the horizontal profile texts were presumed to be horizontally aligned. When the value of the horizontal profile is lower than the HTR/C threshold, it is viewed as the valley splitting the candidate region from the background. Equation (2.2) defines the HTR (horizontal text region), Similarly, Equation (2.3) defines the VTR (vertical text region) and VTR/C is employed for

vertical detecting of the candidate region. For the vertical detecting of the candidate region, the region detected using the horizontal profile is employed. For computation of the VTR and HTR areas, the sum of the edges in the vertical or horizontal orientation is divided by the region size, which estimates the density of the edge down the scan line. The valleys detected surrounding the right and left ends within the vertical profile are regarded as non-text or noise elements.

$$HTR = \frac{sum\ of\ horizantal\ edges}{selected\ region\ size} \qquad (2.2)$$

$$VTR = \frac{Sum\ of\ vertical\ edges}{width\ of\ image} \qquad (2.3)$$

A Samsung Smart-phone was employed to acquire Korean signboard images for experiments which consisted of a total of 445 images. The detection rate was 57% in which instance the background and main text are divisible although noise and isolated regions are included. The suggested technique may additionally be used for vertically aligned texts, although texts were presumed to be horizontally aligned.

An edge-based technique referred to as edge-ray filter was suggested by Huang et al. (Huang, Shivakumara and Uchida, 2013) for detection of the scene character. In this technique, the main function comprised of filtering out complex backgrounds by fully employing the essential spatial format of edges instead of the supposition of straight text line. Extraction of edges is done using a mixture of EPSF (Edge Preserving Smoothing Filter) and Canny. A new EQCA (Edge Quasi-Connectivity Analysis) is employed for the unification of complex edges in addition to contouring of broken character. Redundant rays and non-character edges are subsequently filtered out using LHA (Label Histogram Analysis) by establishing suitable thresholds. Finally, through the exploitation of two regularly employed heuristic rules, occupation and aspect ratio, obvious false alarms are rejected. The suggested technique can encompass bright-on-dark as well as dark-on-bright characters simultaneously, and precise character segmentation masks provide the capability of addressing special circumstances. Experiments are carried out on the benchmark ICDAR 2011 Robust Reading Competition dataset. The results showed detection accuracy was 62.86 for F measure.

Texture features: This technique deals with texts as a unique kind of texture and use their textural characteristics. This implies that wavelet coefficients, Fourier transform, Local Binary Pattern (LBP), filter responses, local intensities and Discrete Cosine Transform (DCT) are

employed to differentiate between non-text and text areas within the images. These techniques are normally computationally costly that due to the all scales and locations requiring scanning. Furthermore, these techniques mainly deal with horizontal texts and are sensitive to scale change and rotation. Although these methods are stronger than the CC methods in addressing complex background, the key disadvantage is their complexity (Zhu, Yao and Bai, 2016)(Ye and Doermann, 2015)(Gllavata, Ewerth and Freisleben, 2004) .

To detect texts in images, a novel texture-based technique was employed by Kim, et al (2003)(Kim, Jung and Kim, 2003). The textural properties (intensities of pixels) of texts are analyzed using a support vector machine (SVM). Then text regions are identified through the application of a continuously adaptive mean shift algorithm (CAMSHIFT) to the outcomes of the texture analysis. Strong and efficient text detection is produced from the combination of SVMs and CAMSHIFT. This method has the ability to detect text from complex images but faced problem in detecting small and low contrast text.

A text localization algorithm system is suggested by Gupta and Banga, (2012), which is designed to locate text in various types of images. The input image is decomposed by Haar discrete wavelet transform DWT into four sub image coefficients. For the three detailed components the Sobel edge detector is applied, and the obtained edge are integrated to create an edge map. The morphological dilation is carried out on the binary edge map and the connected components are marked as text regions. The text is finally extracted within a bounding box based some particular condition. Reflection and illumination effects, image intensity and color variations do not affect the algorithm. Only text boxes and multiple characters are analyzed by this algorithm. The technique is quite computationally efficient. (Gupta and Banga, 2012).

A methodology for the detection and extraction of text regions from low resolution natural scene images was suggested by Angadi, S.A. & Kodabagi, M.M., (2009). The suggested work employs DCT based high pass filter for the extraction of constant background and is texture-based. On each 50 x 50 block from the processed image the texture features are acquired, and with the use of newly detailed discriminant functions which classify each block into two classes. Based on the classes of feature vectors, possible text blocks are identified. The first class of features relates to text blocks while the second relates to non-text blocks. Two thresholds are employed by the discriminant functions: T1 initialized to 0.4 and T2 to 50 relatives to homogeneity and contrast values respectively. These values for T1 and T2 comprise

heuristics selected with basis on experiments carried out on a number of varying images. In addition, to extracted text regions, the detected text blocks are combined and refined. The block schematic diagram of the suggested model is provided in Figure2.5 below.

Good results have been produced by the suggested methodology for natural scene images which containing texts of varying alignment, font and size, with different backgrounds. Additionally, the technique detects regions of nonlinear text. Nonetheless, it detects text regions that are significantly greater than the actual size when the image background is unusually complex and containing automobiles, trees and additional features from outdoor scene sources for some images, (Angadi and Kodabagi, 2009).

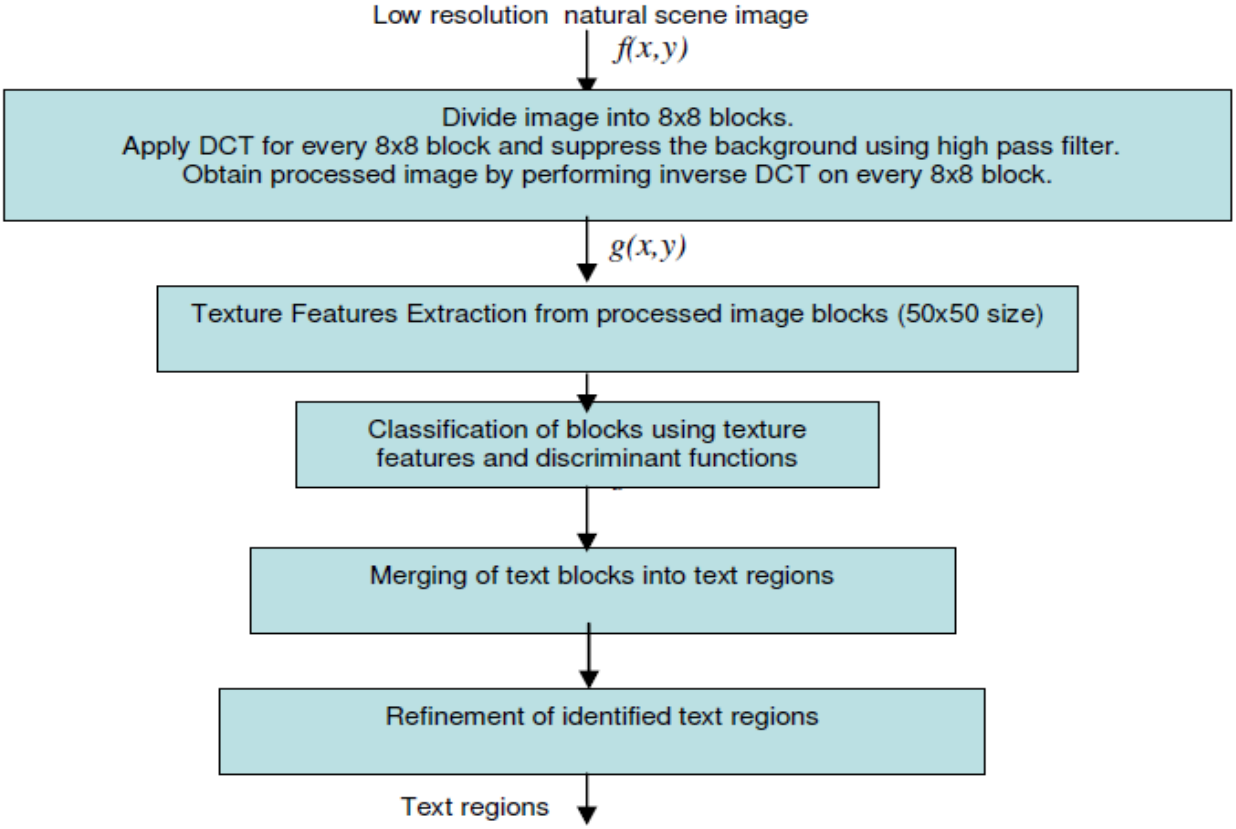


Figure 2.5: Flowchart of proposed method (Angadi and Kodabagi, 2009)

Conditional random field (CRF) has been employed by Zhang et al. (2011) to provide connected components with 'text' or 'non-text' labels. The origin of this paper is the algorithm suggested by Pan, et al , (2011) which employed a CRF model as well and illustrated that text-like background regions are recognized as text characters. In this paper, the authors suggested a two-step iterative CRF algorithm comprising a Belief Propagation inference phase as well as an OCR filtration phase. Within the respective iterations, two kinds of neighborhood

relationship graph are employed for the extraction of multiple text lines and moving of uncertain CCs to the second iteration. A second chance is provided by the second iteration to the uncertain CCs, which filters false positive CCs assisted by OCR. The objective of the suggested technique is the extraction of text lines as opposed to separated words as the ground of truth ICDAR 2005 competition, which results in a lowering of accuracy and recall rate. Zhang et al. method was applied on ICDAR 2005 the result showed that the Precision rate and Recall rate was 56.7% and 56.9% respectively. Zhang et al. method unable to detect blur, curve, individual characters, highlights transparency of the texts (Zhang et al., 2011).

- **Stroke Width Transform (SWT):**

To address the restrictions of the prior techniques, including the challenge in choosing the best features in detection of scene text and raised computational complexity, a new technique referred to as Stroke Width Transform (SWT) is employed. The value of every color pixel is transformed using SWT into the width of a stroke and subsequently bordering pixels comprising approximately a comparable stroke width are combined into connected components. Such a technique is capable of detecting text in spite of direction, scale and font variations. (Zhu, Yao and Bai, 2016)(Ye and Doermann, 2015).

"A stroke is defined as a continues part of an image that forms a band of nearly constant width" (Epshtein, B., 2010). For every pixel the width of the most likely stroke comprising this pixel is computed by a local image operator which is the stroke width transform. As portrayed in Figure 2.6, a stroke is a part of an image which creates a band of almost uniform width. Knowledge of the actual width of the stroke is not presumed; rather, the objective is to recover it. The input image is first transformed into grayscale and an edge detector (Canny Edge detector) is subsequently employed to generate a binary edge map. Stroke width is then computed for every pixel through the detection of parallel lines. Pixels comprising comparable group width are collected to make up a single character. This technique is sensitive to the quality of the retrieved edges which are themselves susceptible to extent of noise and blur within an input image (Epshtein, Ofek and Wexler, 2010).

SWT is a very robust method, the power of the method comes from the ability to show good detection results without the need to feature descriptors. However, SWT unable to deal with some type of alphabets such as the Chinese alphabet. that due to the inconsistency of the Chinese alphabet on contrary to the English alphabet.

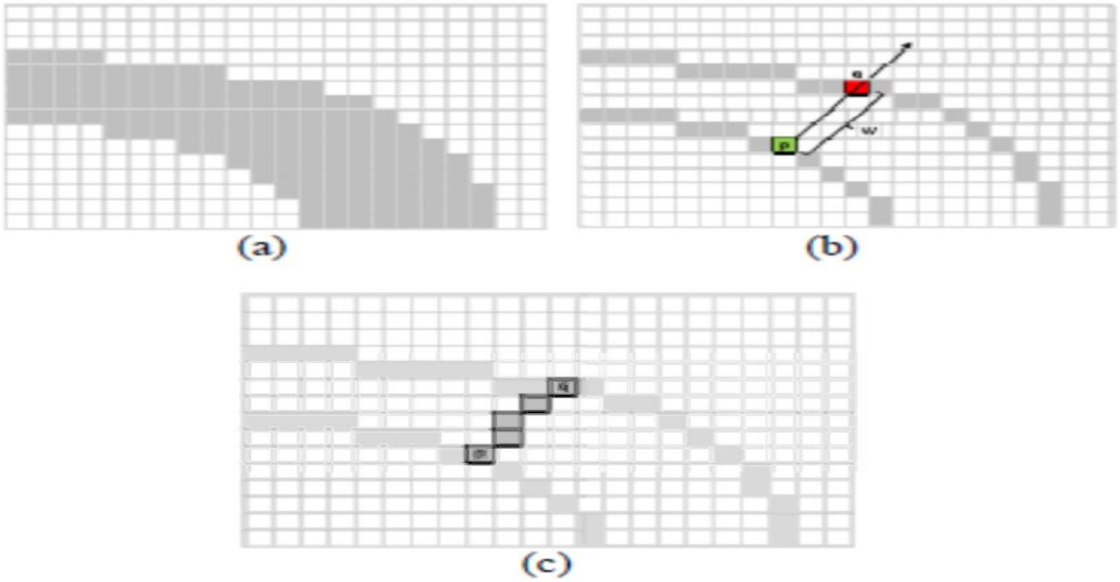


Figure 2.6: (a) A representative stroke. The stoke pixels are darker than pixels of the background. (b) P refer to the pixel that located on the boundary of the stroke. Seeking in the path of the gradient at P, results on reaching q the corresponding pixel on the opposite side of the stroke. (c)The pixels are specified by the minimum of its current value .(Epshtein, Ofek and Wexler, 2010)

For detecting high resolution scene text, stroke-based features have been proved to be encouraging(Epshtein, Ofek and Wexler, 2010) specially in techniques where they are combined with appropriate learning methods such as in (Zhao, Lu and Liao, 2011) and (Chowdhury, Bhattacharya and Parui, 2012) or enhanced with other cues such as edge orientation variance (EOV) and opposite edge pairs (OEPs) (Bai, Yin and Liu, 2012), or combined with spatial-temporal analysis (Mosleh, Bouguila and Hamza, 2012).

(Epshtein, Ofek and Wexler, 2010) introduced a fast method for text detection used the stroke width transform (SWT). The input image is first converted to grayscale and then an edge detector is used to produce a binary edge map. Parallel lines are then detected and used to calculate stroke width for each pixel. Pixels with similar stroke width are grouped together to form a single character where two neighbouring pixels may be grouped together if they have similar stroke width and their SWT ratio does not exceed 3.0 see Figure 2.7. Two most recent text detection competitions: ICDAR 2003 and ICDAR 2005 have been used to evaluate the performance of this method. Compared to the other recently available method, the algorithm reached outperformed all of them. It is capable of detecting texts in several types of fonts and languages.

The method is sensitive to the quality of the extracted edges which in-turn depend on the level of noise and blur in the input image

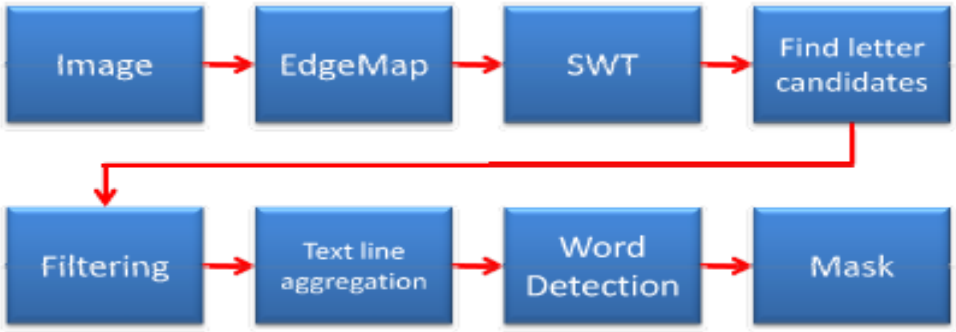


Figure 2.7: The diagram of Epshtein, Ofek and Wexler method (Epshtein, Ofek and Wexler, 2010)

A hybrid algorithm proposed by Zhao, Y., et al, (Zhao, Lu and Liao, 2011) for fast detection of video texts under complex backgrounds, The approach consist of two stages. After segmenting video frames into groups of N×N pieces (N is a scale factor related with video frame resolution) a SVM classifier used to search candidate text-like seed which is then trained by features acquired by a new Stroke unit Connection (SOIC) factor. Each piece is divided into 5 partitioned sub-regions to extract a 25- dimensional SOIC feature see Figure 2.8. SOIC is used to describe the shape distributions of each stroke unit inside a piece with two considerations: unit width and sub-regions connectivity. This approach is used to describe the distribution of stroke inside each over-segmented piece where stroke unit is represented as a text-like segment inside a piece.

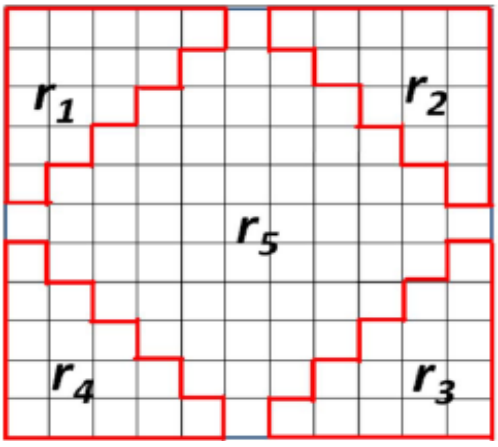


Figure 2.8: Pieces partition with N=11(Zhao, Lu and Liao, 2011)

By using the SONIC factor in the first stage each stroke unit has been well extracted inside each seed piece which make this stage relatively efficient. The experimental results conclude that this method is color and language independent, and robust to video illuminations.

(Bai, Yin and Liu, 2012) produced a robust method to the change of stroke intensity and width which is effective and real time stroke-based for text detection in video. Edge orientation variance (EOV) and an opposite edge pair (OEP) feature are used in this method to characterize the text confidence. Text confidence map (TCM) is a gray-scale image with the same size as the input image. For each pixel, the probabilities of belonging to text region are measured. Then TCM used to detect and group text candidate regions that are become easier than that on the original image. Three features based on the edges considered to generate the TCM are edge density, variance of edge orientation and the number of OEP. see Figure 2.9 OEP feature describe the nature of strokes.

OTSU thresholding algorithm and connected component analysis have been used to extracted and group candidate text components into text lines. Then, boundary box (text box) has been used to enclose each candidate region. Although the text confidence map is fairly accurate, there are a few false alarm blocks. For this reason, those text blocks which are too small to contain text are filtered out by using two constraints as follows:

Min (text _ box _ width, text _ box _ height) < t1 , (2.4)

Max (text _ box _ width, text _ box _ height) < t2, (2.5)

The value of t1 and t2 threshold are predefined and determined by the size of text in the video frame. The performance of the proposed approach was evaluated by using the ASIA-MOVIE database belonging to CASIA-VDB database which includes six video clips containing Chinese, English and a few digital texts. Experimental results demonstrate that the proposed method can detect multilingual texts in video with fairly high accuracy.

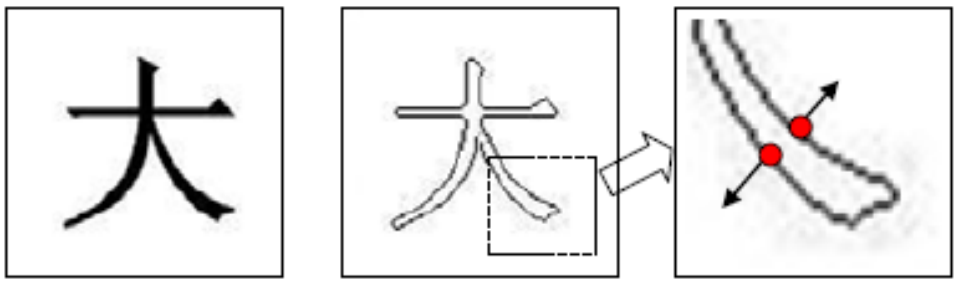


Figure 2.9: diagram of (OEP) method: (a) original image. (b) edge image, (c) (OEP) (red edge points) (bai, yin and liu, 2012)

(Chowdhury, Bhattacharya and Parui, 2012) consider the characteristics of scene text and after an extensive study concluded that: The strokes' thickness bounded by curves may either be uniform or nearly-uniform, the piece of text colour is nearly uniform, and the colour of text and its background is perceptually distinct.

A novel set of features for detection of text in images of natural scenes have been used where the feature vector built depends on the above characteristics. These characteristics which consist of the estimate of the uniformity in stroke thickness using only a subset of the distance transform values of the concerned region. The distance between the foreground and background colours is calculated as a perceptually uniform and illumination-invariant colour space. The two ratios which are anti-parallel edge gradient orientations, a regularity measure between the Canny edge map of the object and skeletal representation. Variation in the foreground grey levels and average edge gradient magnitude are the remaining features of feature vector. Then a multi-layer perceptron (MLP) have been used as classifier. They evaluate the results by using the ICDAR 2003 dataset and another dataset of scene image consisting of text of Indian scripts.

Bandlet-based edge detector has been used recently by (Mosleh, Bouguila and Hamza, 2012) to improved SWT. This enhances text edges as well as rejects noisy and is thus applicable to low resolution texts. A feature vector created from connected components produced via the stroke width transform is used. The feature vector consists of the features established by the stroke width transform and other features such as high contrast with background, variant directionality of gradient of text edges, and geometric properties of text components. To recognize text and non-text components k-means clustering has been employed on the feature vectors. Text components, which are gained from previous step, are grouped and the remaining components are rejected. The technique was evaluated on the ICDAR text locating contest dataset and indicated a considerably improved performance for both the proposed edge detector and the text detection scheme and proved that bandlet-based edge detector is quite effective at obtaining text edges in images while dismissing noisy and foliage edges.

- **Maximally Stable Extremal Regions (MSER)**

MSER (maximally stable extremal regions) is employed within computer vision as a technique of blob detection in images. (Matas et al., 2004) suggested this technique to establish relationships among elements of images comprising various perspectives. This technique of

retrieving an inclusive amount of related image elements has resulted in improved stereo matching, as well as objects recognition algorithms, and contributes to the wide-baseline matching (Matas et al., 2004)(Zhu, Yao and Bai, 2016).

Recently, approaches based on the external-region(ER),which belong to the CC-based methods, won the first places in both ICDAR-2011and ICDAR-2013 competitions and its variants have been widely used to solve text detection problem (Yin, Huang and Hao, 2013).

The key benefit of MSER-based techniques as opposed to conventional component-based techniques is robust in the employment of the character extraction MSERs algorithm, as even in the presence of low quality (low contrast, strong noises, low resolution) the MSERs algorithm can detect the majority of characters, although regardless of their advanced performance a number of open challenges require addressing.

First, the definition of extremal-region is not observed strictly by some of the text objects in images, thus, the whole region's pixels minimum (maximum) intensity value is greater (smaller) than the boundary pixels' maximum (minimum) intensity value. For instance, in Figure 2.10 (b)(c) the characters within red bounding boxes and the grey region in Figure 2.10 (a) would not be extracted as ERs. Even though this is not a serious challenge in ICDAR datasets with over 94 % of the text objects as external-regions, extra consideration must be given to other circumstances (Sun et al., 2014, 2015).

Figure 2.10: Restrictions of E-R based methods (Sun et al., 2015).

Secondly, as Figure 2.11 besides text components, ER methods are also able to extract numerous non-text components, as well as numerous ambiguous components. Ambiguity means that it is unclear or confusing. This ambiguity can involve of

(1) The component in itself is ambiguous, as shown in Figure 2.11(a) and

(2) The forms of some non-text components are considerably similar to characters Figure 2.11 (b). Figure 2.11 (c) demonstrates the distribution of the two types of ambiguous samples.

This ambiguity challenge will become worsen when the components are represented using handcrafted features (the features that are extracted manually, where a set of features identified and extracted such as edge detection, corner detection, histograms) due to the fact that some non-text components use comparable properties as text (Neumann and Matas, 2011, 2012)(Shi, Wang, Xiao, Zhang and Gao, 2013). For instance, uniform stock width. Other CC-based methods also have the ambiguity problem observed in ER-based methods, (for instance, (Epshtein, Ofek and Wexler, 2010)(Yao et al., 2012). Earlier techniques mostly employ line information, including grouping CCs into candidate text lines in order to reduce ambiguity. Nonetheless, resulting from two reasons, this is still a challenge:

(1) Some non-text lines could have the same textures or properties as text lines shown in Figure 2.12.

(2) Some text lines just comprise a single character. When it comes to text and non-text classification

The ambiguity problem is the bottleneck; it has a huge effect on the performance of the text detection algorithm, and hence, requires appropriate attention.

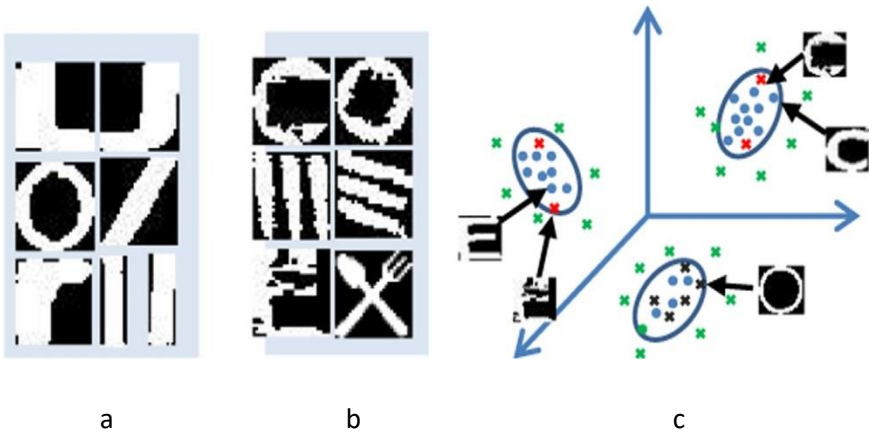


Figure 2.11: Ambiguous samples and their distribution for illustrative purposes

Figure 2.12: Instance of text-like textures

Thirdly, another great challenge is building the candidate text line, mostly if background or layout is complicated. Merging some neighboring non-text CCs to a text line is not so complex and easy, particularly on both line ends. Furthermore, the recurring components in the hierarchical structure as illustrated in Figure 2.13 and also discussed in (Yin, Huang and Hao, 2013), have a severe effect on the candidate text-line formation. (Yin, Huang and Hao, 2013) (Sun et al., 2014, 2015).

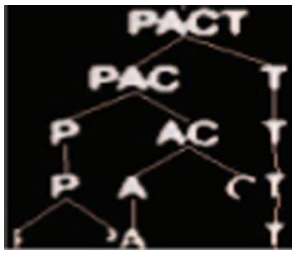


Figure 2.13: The duplicate of MSER

In order to address the above problems, a technique based on generalized color enhanced contrasting extremal region (CER) and neutral networks was suggested by (Sun et al., 2015). Provided a color natural scene image, six component trees are constructed from its grayscale image, hue and saturation channel images in perspective-based illumination invariant color space, with their inverted images, correspondingly. Generalized color-enhanced CERs are extracted as character candidates out of every component tree. Through the employment of a "divide and conquer" approach, every candidate image patch is marked consistently by rules as one from the following five categories; Long, Thin, Fill, Square-large and Square-small, and categorized as non-text or text by a related neural network. Following pruning of non-text components, recurring components in every component-tree are pruned through the employment of area information and color to acquire a component graph, out of which candidate text-lines are created and confirmed by an additional set of neural networks. In conclusion, findings from six component-trees are merged and a post-processing step is employed to recover lost characters and divide text-lines into words as necessary (see Figure 2.14). The method attains 87.03% accuracy, 85.72% recall and 86.37% F-score on ICDAR-2013 "Reading Text in Scene Images" test set.

Canny edges with MSER are combined by (Chen et al., 2011) in order to assist with the weakness of MSER blur. Firstly, MSER is applied to the image to establish the character regions. All the pixels on the outer surface of the boundaries formed by Canny edges are deleted, in order to enhance MSER. MSER's usability in the extraction of blurred text is greatly

enhanced by the division of the letter displayed by the edges. In addition, generation of the stroke width transform image of the regions has been suggested, through the employment of the distance transform for effective acquisition of more dependable outcomes. Pairing and filtering of CCs is performed through the application of the stroke and geometric width information. Finally, the letters are grouped into lines through classification of inter letter distances into two classes: word and character spacings. The distances separating the vertical projections of every character along the text line are calculated, including performing a two class clusterisation through the employment of Otsu's method, including further checks which are done to eliminate false positives.

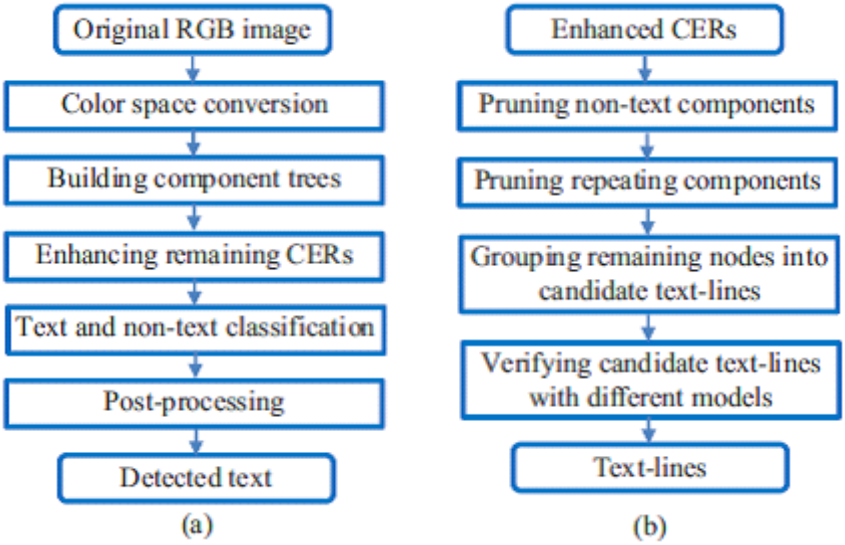


Figure 2.14: Flowchart of (Sun et al., 2015) method: (a) the system; (b) steps of the classification of text and non-text module.

Shi, C. et al (Shi, Wang, Xiao, Zhang and Gao, 2013) employed MSER's text detection approach through the use of graph models. To produce preliminary regions, the technique applies MSER to the image. These are employed in the subsequent construction of a graph model, with emphasis on the position and color distances separating every MSER, which is regarded as a node. Subsequently the nodes are divided into foreground and background with the use of cost functions. A cost function comprises correlation of the distance separating the node and the foreground or the background. The cost function penalizes nodes for its considerable diversity from its neighbor. The graph is cut to split the text nodes from the non-text nodes after they are minimized.

Neumann, L. &Matas, J., (Neumann and Matas, 2011) employed the MSER algorithm in various projections in order to allow text detection in a general scene. Green, blue and red color

channels in addition to the grey scale intensity projection are employed to detect text regions with distinct colors, but not actually distinguished in grey scale intensity. An average Support Vector Machine (SVM) classifier with Radial Basis Function (RBF) kernel was employed. Manual annotation of MSERs retrieved from real-world images downloaded off Flickr was employed to obtain a set of 1227 characters and 1396 non-characters on which a classifier was trained. There was 5.6% classification error by cross-validation. The set used for training is comparatively minor and surely does not include all feasible fonts, characters or even scripts, and no great enhancements in the classification accomplishment rate were produced through lengthening the training set with further illustrations. This shows that features employed by the character classifier are not sensitive to alphabets and fonts.

Hybrid Methods

Different classes of text have different characteristics. For instance, intensive characters and strong gradients are some characteristics of video captions. By contrast other characters could be sparse but color differentiates them from their circumference. Hybrid features are utilized in in detecting such text to enhance the robustness of various text categories (Zhu et al. 2016)(Ye & Doermann 2015).

Pan, Y., Hou, X. & Liu, C., (Pan, Hou and Liu, 2011) found that techniques which are region-based and CC-based are complementary, encouraged by the work integrating global and local information to deal with tasks of object recognition and detection. On their own, CC-based or region-based techniques are incapable of sufficient detection and localization of text. In particular, local texture information can be extracted to segment candidate components accurately by region-based techniques, while filtering of non-text components and accurate localizing of text regions is done by CC-based techniques. Therefore, through integration of CC-based and region-based information, a hybrid text detection and localization technique are achieved.

Pan, Y., Hou, X. & Liu, C., 2011; Pan, Y., Hou, X. & Liu, C. (2008) suggested a three stage system. In the pre-processing stage a text region detector detected text areas within every layer of the image pyramid, then the text confidence and scale information are projected back to the initial image. Subsequently, scale-adaptive local binarization is implemented to create the candidate text components see Figure 2.15 where the high probability represented by red color while low probability represented by blue color.

A CRF model integrating unary component features (encompassing the text confidence) and binary contextual component relationships is employed in the filtering of non-text components during the connected component analysis (CCA) phase. adjacent text components are connected using a learning-based MST (minimum spanning tree) algorithm, and an energy reduction model is employed for cutting off word edges/between-lines to divide text components into words or text lines during the final phase. Good performance is obtained by this technique although it is complicated through the inclusion of post-processing phase following minimum spanning tree clustering (Pan, Hou and Liu, 2008, 2011).

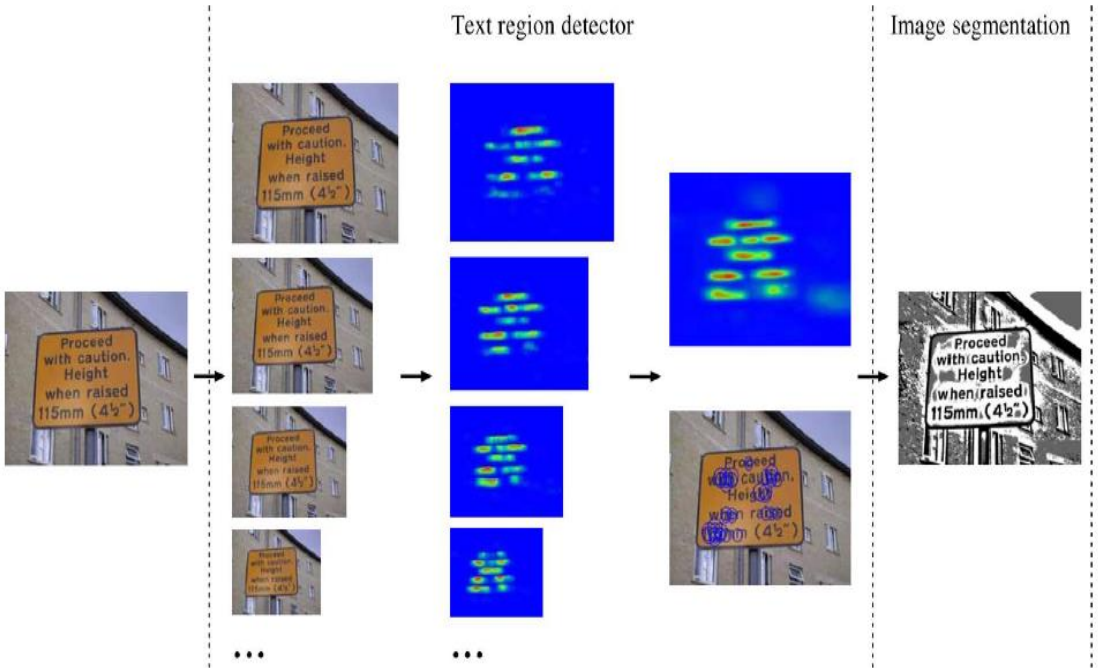


Figure 2.15: pre-processing stage, (a) Original image, (b) Image pyramid and corresponding text confidence maps, (c) Final text confidence map and text scale map, (d) Binarized image

A technique for text region localization and non-text region removal was suggested by (Seeri, Pujari and Hiremath, 2015) for natural scene images with different backgrounds. This hybrid technique for localization and detection of text employs wavelet-based features from edge images relating to input natural scene images to locate text. Haar discrete wavelet transform is employed in the suggested technique, by which the image is decomposed into four sub-bands or components, three detail components (H,V,D) and one average component. The sub-bands of detail components are employed in the detection of candidate text edges within the initial image. Fusion of the edge information comprised within the three sub-bands is made

available by discovering the edges within the three sub-bands, specifically diagonal (D), horizontal (H) and vertical (V) sub-bands.

Due to its effectiveness in locating strong edges relevant to text, Sobel edge detector is employed. The subsequent stage comprises creation of an integrated edge map employing logical “AND” and “OR” operators, which extracts some fuzzy thresholding, non-text regions as well as the morphological operators for the classification and segmentation of the text regions. Accuracy and recall rates of 79.54% and 89.21% respectively have been provided by the suggested technique. The efficiency and robustness of the suggested technique for localization and detection of multilingual text areas are showed by the experimental outcomes, with variations in lighting and orientation of text within an image, font size, font style and scale.

A technique for text region localisation and non-text region removal was suggested by Seeri, Pujari and Hiremath (2015) for natural scene images with different backgrounds. This hybrid technique for localisation and detection of text employs wavelet-based features from edge images. Relating to inputted natural scene images to locate text, Haar discrete wavelet transform is employed in the suggested technique, whereby the image is decomposed into four sub-bands or components, namely three detail components (H, V, D), and one average component. The sub-bands of detail components are employed in the detection of candidate text edges within the initial image. Fusion of the edge information comprised within the three sub-bands is made available by discovering the edges within the three sub-bands, specifically the diagonal (D), horizontal (H) and vertical (V) sub-bands.

2- Text Recognition

Text recognition is the process of converting image regions into a sequence of characters as strings of texts (Zhu et al., 2016)(Ye & Doermann, 2015).

- Character recognition

Character-based techniques carry out recognition at the level of characters which are the building blocks of text, and because of this they contain a considerable amount of significant information. Effective character recognition results in simpler application of bottom-up text (Liu, Meng & Pan, 2019).

Yao et al. (2014) explained new means of representation used for characters recognition that pay greater attention to the sub-structures features of characters, including arc, bar and

corner of entire characters. These are basically primitive characters that are termed Strokelets. Strokelets can differentiate between different characters as well as between the characters and their background. Yao et al. identified a total of 62 classes consisting of 52 English letters and 10 Arabic numbers by employing the Bag-of-Strokelets, which is composed of a histogram of binning Strokelets and Random Forest. This technique carried an accuracy of 80.33% for ICDAR 2003 (FULL), 88.48% for ICDAR 2003 (50) and 75.89% for SVT. The advantages of this technique are robustness to distortion and its generality to different languages.

The characters were illustrated by Shi et al. (2013) using a tree-structure, where the nodes represent character parts, as illustrated in Figure 2.16. The resultant tree combined each character's global structure and its local appearance. This was followed by computation of the CRF model for each possible position and the outcome of the recognition was subsequently computed by employing the function of cost (the latter computation depended on the detection values, spatial restrictions and knowledge of language).

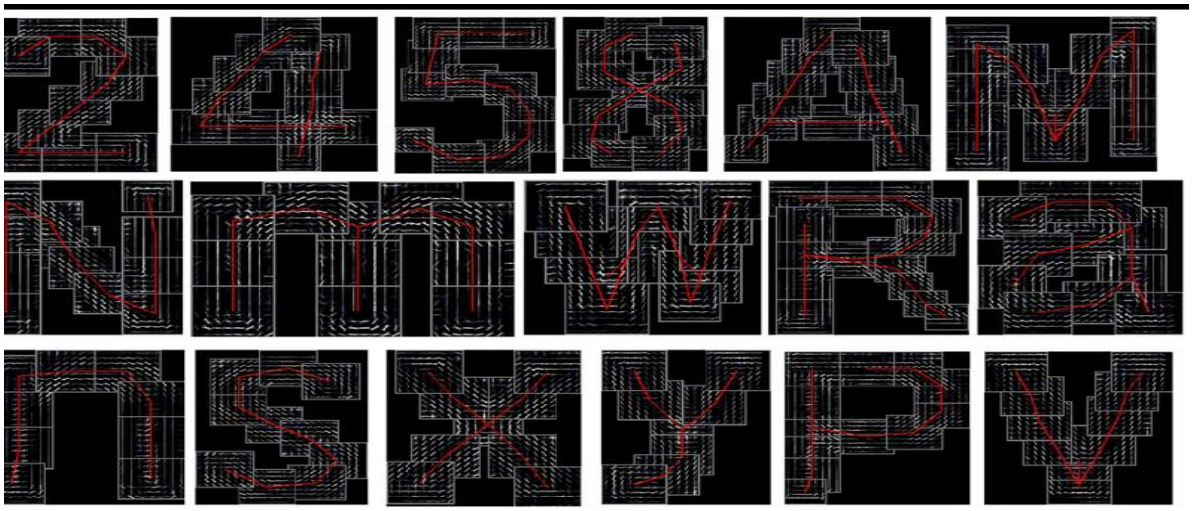


Figure 2.16: The tree-based model of structure for several characters as recommended by Shi et al. (2013). Topological associations of one part are indicated by red, whereas one rectangle equates to a filter that's founded on a part (Shi, Wang, Xiao, Zhang & Gao, 2013).

Groups of samples from ICDAR 2003, ICDAR 2011 and SVT were used to examine the recommended techniques, while the findings indicated that the recognition accuracy levels were 79.30%, 82.87%, and 72.51%, respectively (Shi, Wang, Xiao)(Zhang & Gao, 2013) .

Blur, noise, partial occlusion and different fonts of texts could all be recognised by the recommended technique.

Lee et al.'s study focused on character recognition and demonstrated a discriminative features pooling technique (Lee et al., 2014). The initial step involved the extraction of features

at the lower level, including colour, and the magnitude of a gradient and gradient histograms. The second step involved automatic combining of these features using a region-based pooling methodology. Finally, the features that were most instructive and useful were ordered and chosen using a linear SVM-based classifier. Comprehensive tests were performed on ICDAR2003, ICDAR2011 and SVT, with findings of 0.76%, 0.77% and 0.80%, respectively.

Lou et al. (2016) developed a progressive shape model employed for scenes text recognition. To be able to manage problematic cases where fonts vary, a greedy strategy was employed for choosing fonts, and said fonts were used to create a graphical illustration from clean font images (please see Figure 2.17). The representation of image level features depends solely on edges, which excludes the effect of other factors, including texture and colour. To be able to identify the edges, 16 oriented filters were utilised to detect edge. The authors then also used local suppression, which maintains a maximum of one edge oriented for each pixel (see Figure 2.17). By choosing one edge each time, landmark features are produced. Moreover, for the results, they selected max margin structured output learning to train a parsing model. This technique, nevertheless, had its drawbacks, as when overexposure or blurry images occurred, edge evidence of the edge were missed

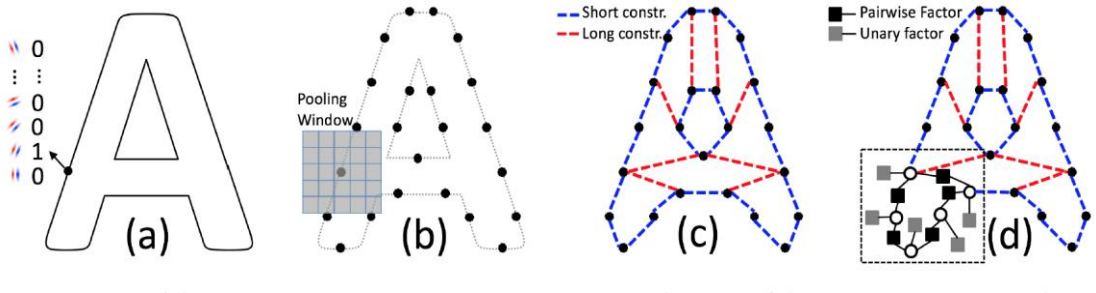


Figure 2.17: The 16 oriented edge features that were identified per pixel.

(a). An example of a process performed that chooses "landmark" qualities out of a compressed edge map (16 oriented edge per pixel) (b) followed by the addition of a pooling window (b). The restriction of the shape model by the addition of lateral constraints (c). A partial depiction of the factor graph illustration of the model (d) (Lou et al., 2016).

Evaluation of the proposed method on the SVT and ICDAR2013 datasets showed an acceptable result of 80.7% and 82.6% respectively.

- **Word Recognition**

Many researchers are also interested in examining the word level, which utilises word-based techniques for recognition.

Feild and Learned-Miller's study recommended the use of a process that utilises open vocabulary word recognition, and which is not dependent on words that come from a particular lexicon (Feild & Learned-Miller, 2013). They created a features vector containing a HOG descriptor per character, as well as a case feature, which is a percentage value that is calculated as a character height percentage as part of the height of the largest character forming the same word. To accomplish this, a linear-chain CRF was used to represent the sequence of characters in a word. CRF was then utilised to identify the primary text label per input image. This was in place of evaluating the probability of each lexicon word followed by selecting the word with the highest probability, as this would have been costly for sizeable web-based lexicons consisting of approximately 13.5 million words.

Then, using global language information, a step was employed for errors correction by assessing the probability for occurrence of lexicon words that consist of no more than two characters from the particular label, as illustrated in Figure 2.18.

ICDAR 2003 and ICDAR 2011 were used for assessing the process and the findings of the tests indicated that recognition accuracies using the open vocabulary word recognition were 62.76% and 48.86 %, respectively.

Goel et al. (2013) attempted to resolve the issue by using a retrieval structure where the lexicon is converted into a collection of synthetics of a complete word image followed by presenting the recognition problem as the means of retrieving the optimum match from the image group of the lexicon. HOG was then employed to extract the features from the image using a group of histograms followed by using k-nearest neighbour. To test and train this technique, the Street View Text (SVT) and ICDAR 2003 datasets were used with recognition accuracies of 77.28% and 89.69%, respectively.

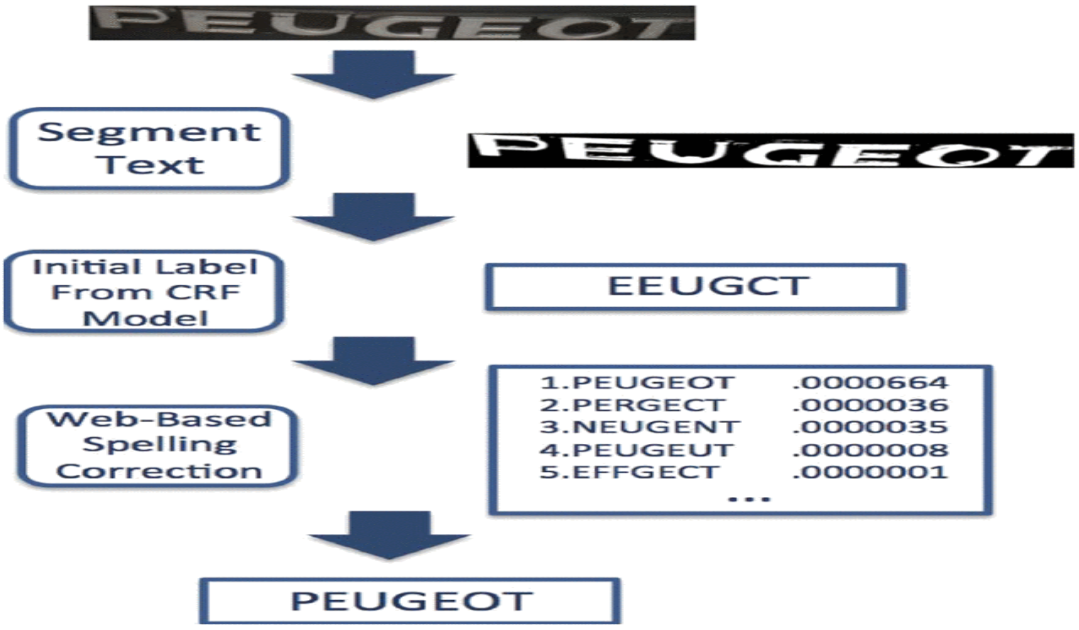


Figure 2.18: An image was split into text foreground and background. A CRF (conditional random field) model was then used to identify the most probable text string by testing the connected components in the segmentation process (Feild & Learned-Miller, 2013).

3- End-to-End systems

The very first end-to-end text detection and recognition was suggested by Neumann and Matas (2011), as illustrated in Figure 2.19. Here, candidate regions were initially extracted using MSER. This was followed by a step to remove regions of non-text by training SVM with Radial Basis Function (RBF) kernel on the ICDAR 2003. The results were identified as characters measuring 35 * 35 pixels. These were then considered as the input data used in the recognition stage, where each MSER mask was sub-sampled into smaller samples in a matrix measuring 5 * 5 pixels. This was carried out to identify the chain-code bitmap for 8 directions. A total of 200 features were then produced using 25 features * 8 directions, and in this case, for the classification stage, SVM was employed. For the training stage, Neumann et al. 2011 used the Windows OS characters. They then employed the Char74k dataset and the ICDAR 2003 dataset as testing sets to evaluate the recognition performance, and this yielded rates of 72% and 67.0%, respectively. The limitation of this technique was that several of the individual characters could not be identified as MSER regions. This was together with the incorrect text line creation or wrong word segmentation.

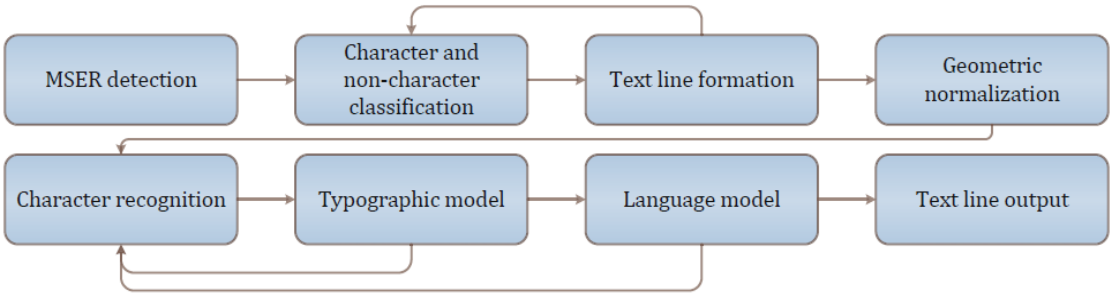


Figure 2.19: Stages of Neumann and Matas (2011) method.

Yao, Bai and Liu (2014) demonstrated that detection and recognition of text could occur simultaneously by utilising the same features and classifiers in a joint framework (Figure 2.20). Clustering and SWT were subsequently employed for identification of text candidates. The clustering algorithm included a greedy hierarchical agglomerative, which was employed to group connected components into chains. Subsequently, the technique used two levels of features and both levels utilised the Random Forest classifier. The first features level was a component level and included use of features such as contour shape, density, occupation ratio, axial ratio, edge shape and width variation. The other level was that of chain features, which included the mean probability and mean turning angle, the candidate count, variation of distance and size, as well as a range of mean values for direction bias, axial ratio, density, width variation, colour, in addition to self-similarity and the mean structure self-similarity.

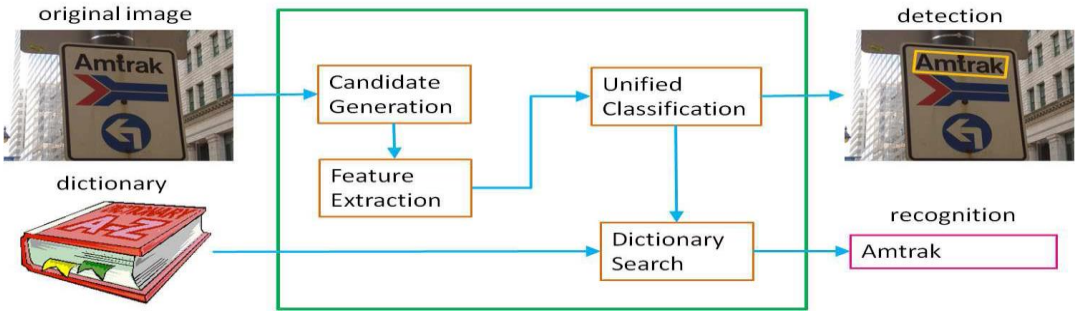


Figure 2.20: Overview of the Yao, Bai and Liu (2014) system.

A dictionary-based search technique was used to achieve greater accuracy with recognition. Yao, Bai and Liu (2014) created a dictionary for the proposed method by employing the top 100,000 most used words on the search engine. Any text detected was split into individual words which were then matched to the elements of the dictionary to identify the corresponding word. 100,000 synthesised images were employed for training. The testing on the dataset ICDAR 2011 yielded a precision of 0.822%, recall of 0.657 % and F-measure of 0.730%.

Moreover, it achieved 0.64% and 0.62%, and 0.61%, respectively on MSRA-TD500, and regarding recognition on ICDAR 2011, it achieved 0.52%, 0.47% and 0.48%.

2.3.2. Scene Text Detection and Recognition Using Deep learning:

Given the advancement of deep learning, the CNN (convolutional neural network) was extensively examined. It carries the benefits of being impervious to deformation, illumination and geometric transformation. The computational cost involved in extracting information immediately from an image is minimal. The CNN has such developed parameters that the function of detection and recognition of text is massively enhanced in the natural scene. Current techniques that are founded on use of the CNN can be generally classified into a number of sub-categories including techniques based on segmentation, region proposal-based techniques and hybrid techniques that utilise multi-task learning. The next sections include discussion of text detection and recognition using deep learning (Long, He & Ya, 2018)(Liu, Meng & Pan, 2019).

1- Text Detection

SWT and MSER are two early algorithms used prior to deep learning, which have had a great impact on the techniques developed post-deep learning.

A technique termed the region proposal-based technique was recommended by Huang et al. (2014). This technique utilises deep neural networks while benefitting from other techniques that are reliant on regions. Using the CNN, these techniques can powerfully learn a representation of a component. They can also take advantage of the CNN's robustness in classifying and detecting objects. Figure 2.21 illustrates the three main phases that constitute the process. The first stage produces components of the text using the MSERs detector on the input image. Following this, a confidence value is assigned to each MSER's component using a trained CNN classifier. To create the definitive text lines, the text parts that contain the higher confidence scores are employed. The methodology of using MSERs decreases the number of searches windows and enhances text detection in cases with low contrasts. The most referenced benchmark is that of ICDAR-2011; this methodology achieves the greatest results, reaching above 78% in F-measure (Huang, Qiao & Tang, 2014).

(a) (b) (c) (d)

Figure 2.21 : The method of Huang, Qiao and Tang (2014) (a) depicts the input image. Initially, the MSERs operator was employed on the input image (b) text component candidates indicated. (c) On implementation of the CNN classifier, the component confidence map was generated. Lastly, a simple thresholding on (d) is generated by the end detection results (Huang, Qiao & Tang, 2014).

SegLink was proposed by Shi et al. (2017). SegLink is a text lines detector whereby lines are detected through the use of two minor elements (segment and link) which are locally detectable. The former (segment) is an oriented box that covers a section of the word, whereas the link is the connection formed between two adjacent segments. Moreover, the pre-training VGG-16 network was used, which consisted of 11 convolutional layers. The outputs were further defined using confidence scores. These are composed of both segments and links that are subsequently joined into complete word bounding boxes. To train, SynthText was used. ICDAR 2015, MSRA-TD500 and ICDAR 2013 were employed to test the results of the recommended techniques, with F-measures of 75.0%, 77% and 85.3%, respectively. SegLink is not able to detect text with large spaces between characters. and thus, both the α and β threshold have to be set manually – α for the segment and β for the link threshold.

DMPNet (Deep Matching Prior Network) was proposed by Liu and Jin (2017). First, quadrilateral sliding windows were employed in a number of particular intermediate convolutional layers to approximately recall text with a higher overlapping region. With regard to this technique, the VGG-16 model constitutes the primary framework of DMPNet, which employs the same complex intermediate layers for implementation of quadrilateral sliding windows. Use of the ICDAR 2015 dataset showed that DMPNet results in an F-measure of 70.64%.

To handle the task of text detection in multiple orientations, languages and fonts, Zhang et al. (2016) integrated the salient map, which is the result of a trained Fully Convolutional Network model named Text-Block FCN, and character components (MSER components) to predict the line text regions. Character-Centroid FCN (a Fully Convolutional Network) was employed to predict each character's centroid which is helped to eliminating false positives, as illustrated in Figure 2.22. This recommended technique was evaluated using three benchmarks

for text detection, including MSRA-TD500, ICDAR2015 and ICDAR2013, with F-measures of 0.74%, 0.54% and 0.83%, respectively.

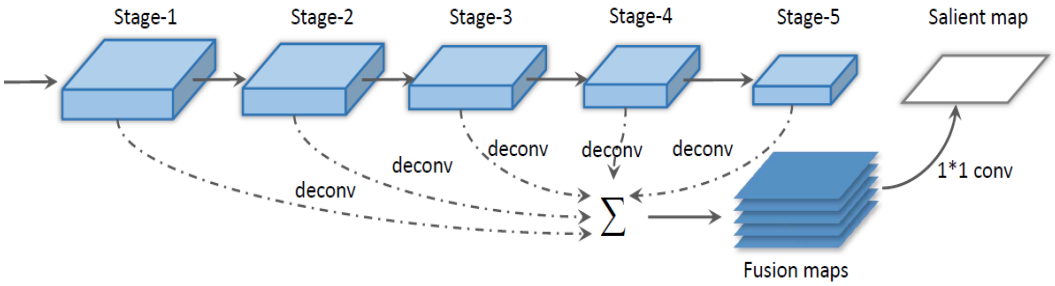


Figure 2.22: An illustration of the Text-Block FCN architectural. Five convolutional layers are found in the VGG 16-layer model. For each stage, a deconvolutional layer is connected. The feature maps are calculated by concatenation (Zhang et al., 2016).

A prediction of the orientation and the generating of text line candidate bounding boxes is possible through the powerful combination of the salient map and the local character components. This technique, nevertheless, has its disadvantages, including slow speed to the extent that it cannot deal with processes in real time. Moreover, there are various factors that relate to the functioning of this process which are not working to optimum levels, including both types of errors, namely false positive and missing characters. These factors also include unbalanced illumination, multiple orientations, dot fonts, broken strokes, perspective alteration and a mixture of multiple languages.

There exist two studies which are closely related in that they both employed FCN and treated detection issues as segmentation issues (He et al., 2017)(Zhang et al., 2016). They differ, however, with regards their management of arbitrary oriented text lines beside curved text lines, which He et al.'s model can manage but Zhang's model cannot. He et al.'s (2017) techniques utilise a new algorithm for text detection that consists of two cascaded steps: the first is FCN, which is a multi-scale fully convolutional neural network, and which is proposed for the extraction of block regions of text. He et al. proposed two networks in a cascaded fashion to solve the problem.

Text areas were extracted using a multi-scale fully convolutional network. The output of FCN acts as the input to the segmentation stage to eliminate false positives. The input may consist of a number of words or lines. When using the word instances in the detection of scene text, the word instances are described as a text line or word which cannot be split off but that remains only visual. To resolve this issue, two networks are recommended that function in a

cascaded fashion: Instance-Aware CNN (IACNN) and Text Line CNN (TL-CNN) (Figure 2.23). The recommended algorithm can precisely localise text lines (including curved ones) or words in arbitrary positions. Note that curved text lines cannot be dealt with in many structures. The algorithm employed in this study resulted in good performance in ICDAR 2013, generating 85%, and in ICDAR 2015, generating 63%, for F-measure. The benchmark datasets used for assessing accuracy were CUTE80 and Street View Text at 78 % and 73%, respectively. This method, however, will fail for particular cases where the text is too blurry, or characters are scattered all over text lines or even low contrast. These all have the potential to cause issues in this process He et al., 2017.

TextSnake expresses text instance as a series of disks that overlap one another (Long et al., 2018). Each disk is related to an orientation and radius and is located at the centre line. TextSnake can change its shape in response to changes in the text instances, including changes in scale, bend and rotation. This study selected VGG-16 to allow for a direct and good comparison with other techniques (Figure 2.24). With regards the feature merging network, several phases are layered in order and each is made up of a merging unit that obtains feature maps from the final phase and from the matched VGG-16 network layer.

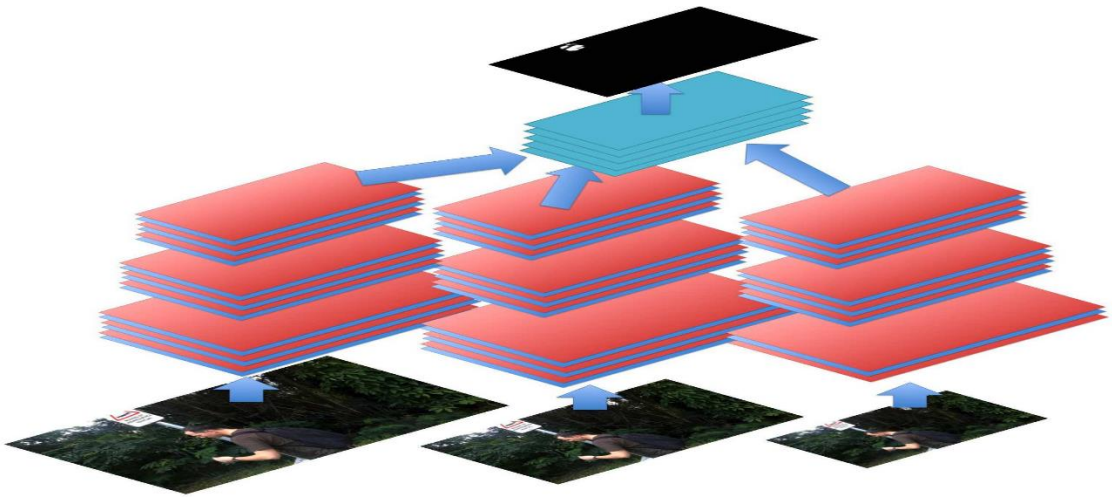
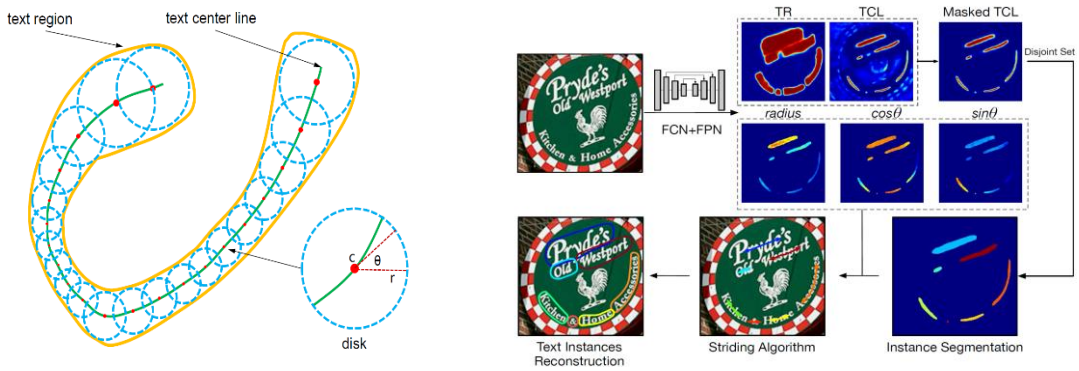


Figure2.23: An illustration of FCN's multi-layer build. The inputs equal 0.5 and 0.25 of the scale used in the input image. Taking account of features from three scales feeds into the forecast process. All three branches share the limits of the convolutional parts (He et al., 2017).



a b

Figure 2.24: (a) Diagram illustrating the suggested TextSnake representation. Yellow denotes the text region which is depicted as a string of disks placed in order as indicated by the blue. Each disk is situated at the centre line which is seen in green. Thus, this is the symmetric axis or skeleton which is related to θ, the orientation, and r, the radius. TextSnake can accurately detail text of various forms, making it more general and flexible, particularly as it ignores lengths and shapes. (b) Illustration of the method framework: including network output and post-processing (Long et al., 2018).

Score maps are predicted by the FCN network. The FCN network predicted the score maps of TCL text centre line, and TR text regions. This occurs jointly with geometry attributes, including r, cosθ and sinθ. As the TCL forms part of the TR, the TCL map is masked by the map of the TR. As TCLs do not overlap each other, a disconnected set is used when carrying out instance segmentation. To extract the lists of the middle axis point, a striding algorithm was employed, following which, as the end step, the text instances were reformed (Figure 2.25). Various datasets were used to assess the recommended technique by computing the F-measure, including Total-Text (74.4%), Curved Text CTW1500 (64.4%), and various kinds of Incidental Scene Text MSRA-TD500 (78.3%) and ICDAR2015 (82.6%).

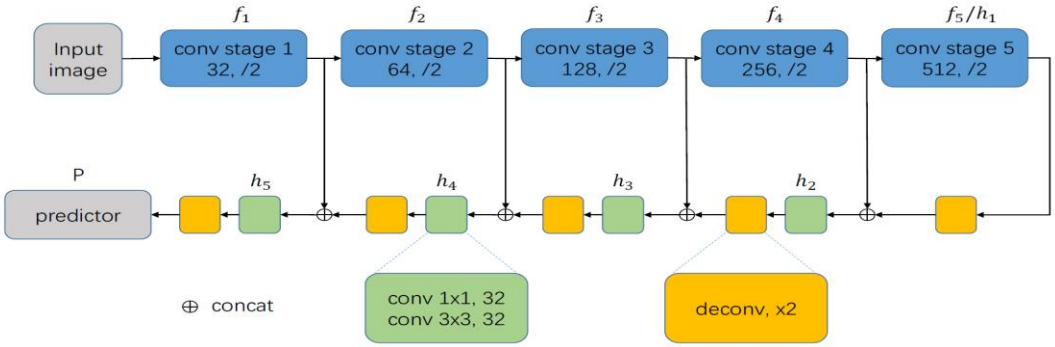


Figure 2.25: Illustration of the network architecture where the VGG-16 convolution stages are represented by blue squares.

The text in this method was treated as a group of local elements instead of a whole item, and these elements were then combined to reach decisions. The recommended flexibility in

presentation is thought to be responsible for the outstanding generalisability. The text is not treated as one item altogether; rather, the representation treats each text as being made up of several local factors which it then combines to reach decisions. Local attributes are maintained when constructed into a full character (Long et al., 2018).

2- Text Recognition

Recently, the area of detection and recognition of scene text has been dominated by deep neural networks.

The CRNN, or Convolutional Recurrent Neural Network, was proposed by Shi et al., and consists of both the RNN and the DCNN (Shi et al., 2016). The CRNN comprises three parts. From bottom to top, these are the convolutional layers, recurrent layers and the transcription layer. Starting from the bottom is the work of the convolutional layers, which is automated and results in the extraction of a feature sequence per input image. The next layer is the recurrent network which was created to predict all frames of the feature sequence that were the outcome of the convolutional layers. The final layer is the transcription layer. This sits at the top of the CRNN and is used to translate the forecasts for each frame output produced by the recurrent layers into a label sequence. A VGG-Very Deep architecture was created, and it forms the framework of the convolutional layers. It was altered slightly to make it appropriate for identifying English text (see Figure 2.26). Performance assessment was conducted using four frequently-employed benchmarks for scene text recognition, specifically IIIT 5k-word (97.6%), Street View Text (96.4%), ICDAR 2003 (91.9%) and ICDAR 2013 (89.6%) when in lexicon-based mode.

An Attentional Scene Text Recognizer with Flexible Rectification, or ASTER, was introduced by Shi et al. (2018). The ASTER utilises a comprehensive modification technique to deal with irregular text issues. The model is illustrated in Figure2.27, and can be seen to consist of two parts: the rectification and the recognition network. The former takes an input image and rectifies it to improve the text it holds. The transformation used is TPS, or parameterised Thin-Plate Spline (TPS). This is highly flexible and can deal with a range of irregular texts. The TPS can improve both perspective and curved text, which are two forms of uneven text. To enlarge the feature context, a multi-layer Bidirectional LSTM (BLSTM) network is used over the feature sequence to make the feature context bigger. The feature sequence is examined in both directions, making it possible to capture long-range dependencies in the two directions. Its output releases a novel feature sequence of the same length.

Six datasets were used to assess their performance, namely IIIT5k, SVT, ICDAR03, ICDAR13, SVTP and CUTE. The outcomes of the two variants are detailed in Table 2.1. From this it is clear that the improved model is superior to the non-transformed one for each dataset, which is very clearly observed in SVTP (+4.7%) and CUTE (+3.1%). As both datasets contain irregular text, a significant impact is seen following rectification.

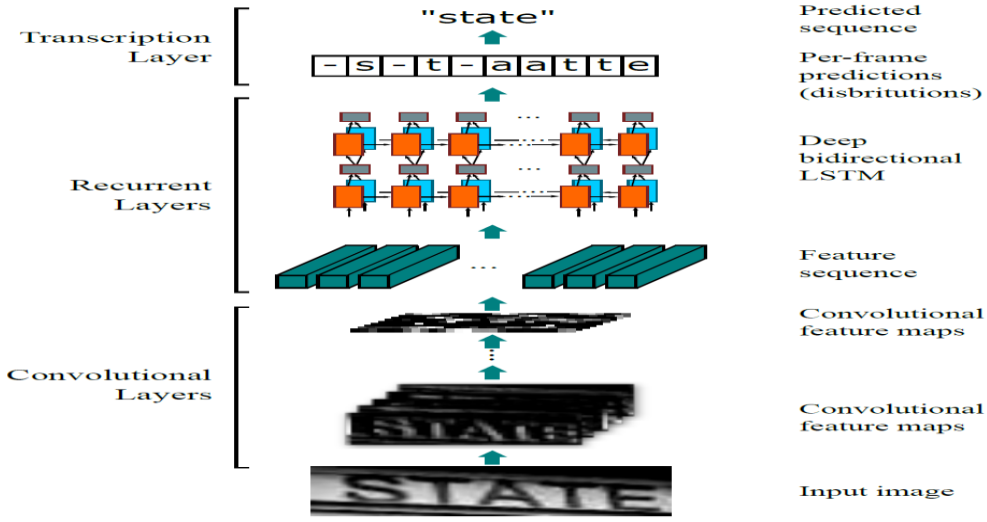


Figure 2.26: An illustration of the three parts that make up the structure of the CRNN, including 1) convolutional layers, tasked with extracting a feature sequence from the image input; 2) recurrent layers, which forecast a label distribution per frame; 3) transcription layer, which translates the per-frame forecasts into the last label sequence (Shi et al., 2016).

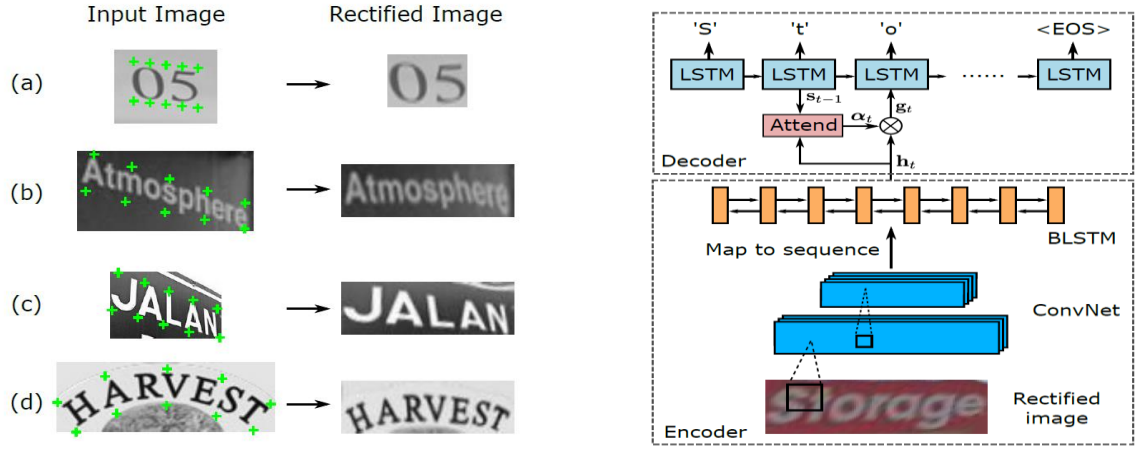


Figure 2.27: Different forms of uneven text can be improved using the TPS transformation, such as loosely bounded text (a), oriented or perceptively distorted (b)(c), and curved text (d), but may include other types as well.

Table 2.1: Accuracies of recognition including and excluding rectification.

Variants	IIIT5k	SVT	IC03	IC13	SVTP	CUTE
Without Rect.	91.93	88.76	93.49	89.75	74.11	73.26
With Rect.	**92.67**	**91.16**	**93.72**	**90.74**	**78.76**	**76.39**

The general CRNN (convolutional recurrent neural network) strategy was recommended by Liu et al. (2018) for use in recognition of scene text in real time. This results from the joining together of the recurrent neural network (RNN) and the convolutional neural network (CNN). The former extracts features, while the latter encodes and decodes sequences of features. The CRNN has been used with the aim of enhancing the accuracy rate of scene text recognition, and various deeper CNN architectures are examined to gain descriptors of the features. In particular, the various deep models are trained by VGG and ResNet, which are also used to gain the images' encoding data. After the extracting of the features of the input image using the ResNet and VGG models, followed by feature encoding into the matching sequence, a BLSTM layer is used to decode the feature sequence. The BLSTM output sequence is then mapped out by CTC to the identified text, as illustrated in Figure 2.28. The efficacy of this technique is exemplified by the text outcomes in public datasets. The results of implementing the proposed techniques on the four datasets were IC03 = 91.5%, IC13 = 88.7%, SVT = 82.8%, and IIIT5K = 82%.

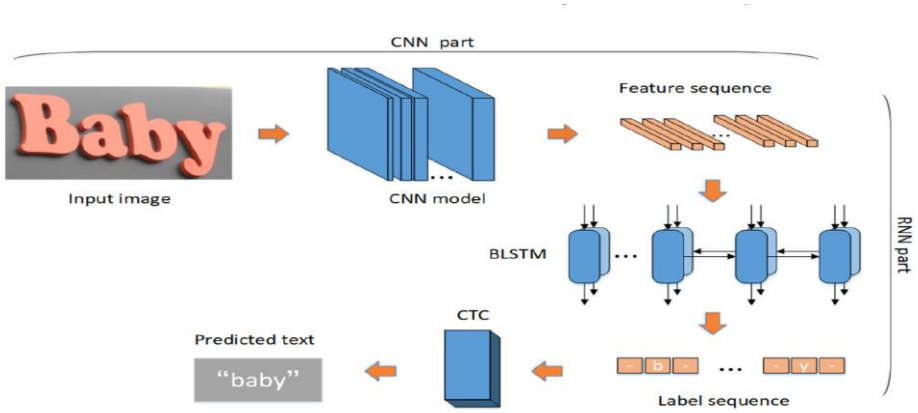


Figure 2.28: Method recommended by Liu et al. (2018). Initially the features are extricated using the CNN, followed by their encoding into the feature sequence. The feature sequence is subsequently decoded via BLSTM and the label sequences are the output. Mapping of the labelled sequence to words by the CTC layer will then take place. Lastly, the output of the recognition step is obtained.

3- End-to-End Systems

Jaderberg et al. (2016) proposed two split steps for detection and recognition systems that function end-to-end. For text detection 'The Edge Boxes region' was a recommended algorithm (Zitnick & Dollár, 2014) which was employed to compute the edge map across many scales and aspect ratios of a sliding window manner. The second was a weak aggregate channel features detector (Dollár & Zitnick, 2015), which was used to aggregate channel features over sliding windows (ACF) for the oriented gradient histogram (six channels). This was followed

by AdaBoost, which was used for classification. Five convolutional layers and three fully connected layers of CNN were utilised for the recognition phase (Figure 2.29). The dictionary of words, composed of approximately 90,000 words, was used for the classification, which was performed at the last completed connected layer. A synthetic data engine was employed to create a dataset comprising 9 million 32 × 100 images (Jaderberg et al., 2014). This dataset had an equal number of word samples obtained from a 90,000-word dictionary. The findings of the performance of the end-to-end on ICDAR 2003, 2011, 2013, and SVT, were 86%, 76%, 76% and 53%, respectively (Jaderberg et al., 2016).

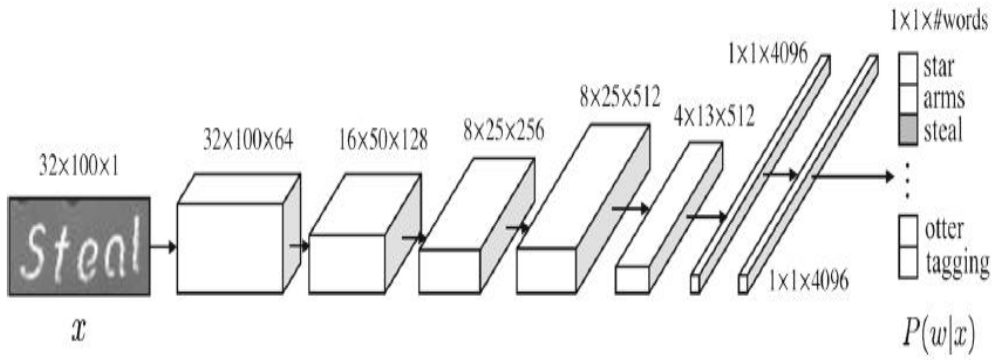


Figure 2.29: A diagram of CNN employed for recognition of text by means of word classification. The diagram clearly indicates each layer of the network's dimensions of the feature maps (Jaderberg et al., 2016).

In contrast with the work of Jaderberg et al. (2016) and Zhang et al. (2016), which consisted of three detection stages, each of which also included more than one algorithm, TextBoxes++ was presented by Liao et al. (2017). This is a word-based scene text detection system which requires the training of one end-to-end network. It involves modifying the final two completely connected layers of VGG-16 to form convolutional layers, followed by the addition of convolutional and pooling layers (conv6 and pool11). As the words usually had large aspect ratios rather than general objects, six aspect ratios for default boxes were indicated. Moreover, five scales were used to change the input image scale. Following this, the CRNN (Shi, Bai & Yao, 2017) was integrated to enable the text recognition method with textboxes and to achieve the end-to-end recognition (Figure 2.30).

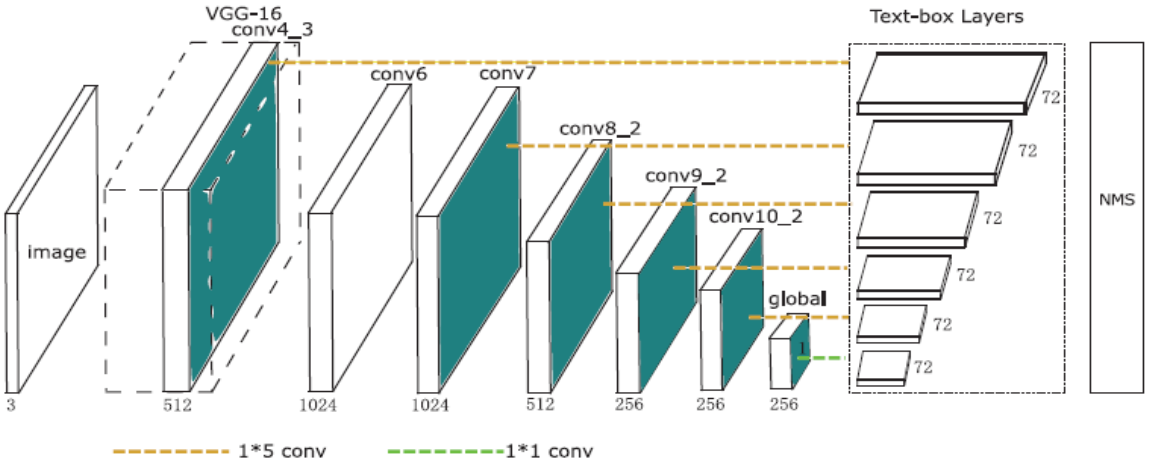


Figure 2.30: Illustration of the architecture of TextBoxes. TextBoxes is a convolutional network consisting of 28 layers. 13 are obtained from VGG-16, and 9 extra convolutional layers are added following the VGG-16 layers. 6 of the convolutional layers have text-box layers attached. A text-box layer forecasts a 72-d vector on each map location. These include for the 12 default boxes the text presence scores (2-d) and offsets (4-d). All the combined outputs of all text-box layers have a non-maximum suppression applied (Liao et al., 2017).

To assess the performance of TextBoxes, the ICDAR 2011 and ICDAR 2013 datasets were used. The results showed that TextBoxes achieved 0.86% for both datasets detection and 0.84% for end-to-end for ICDAR 2013. However, TextBoxes cannot deal with some difficult cases, such as overexposure and large character spacing.

Figure 2.31 illustrates the technique recommended by Lyu et al. (2018). Their technique consists of four components. The backbone is composed of a feature pyramid network (FPN), text proposal generation through a RPN (region proposal network), using a Fast R-CNN for bounding boxes regression, and, lastly, when performing character and text instance segmentation, a mask branch is used. RPN initially produces many text proposals in the training stage. This is followed by feeding RoI features into the mask branch and the Fast R-CNN branch. This results in the production of accurate text candidate boxes, the text instance segmentation maps, and the character segmentation maps. Lastly, the prediction maps then produce the text instance bounding boxes and sequences.

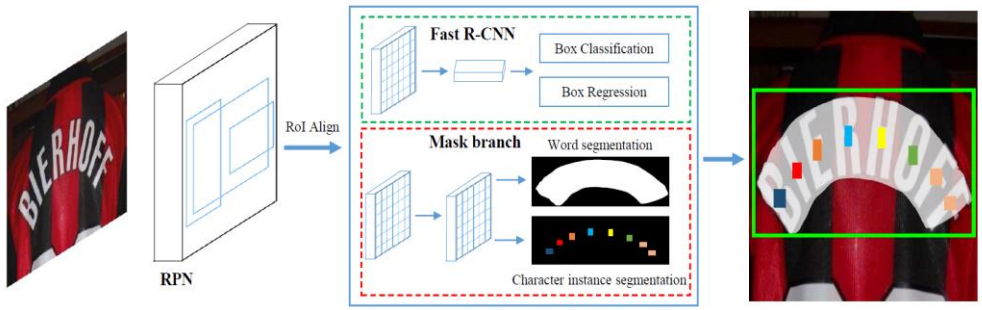


Figure 2.31: Diagram depicting the Lyu et al. (2018) method architecture.

By performing various tests and assessing the datasets using F-measures, the following results were achieved for each as follows: for the horizontal text set ICDAR 2013 gave 86.5%, for the oriented text set ICDAR2015 gave 62.4%, and for the curved text set Total-Text gave 71.8%.

2.4 Measurement for Text Detection and Recognition

Datasets provide ground truth with the images for the texts that are available in the images. The performance comparison of different algorithms usually refers to their precision, recall and F-measure score. To compute these performance indicators, a list of predicted text instances should be matched to the ground.

Mosleh et al. (2012) defined precision as the number of correct estimates divided by the total number of estimates. A method has low precision if the number of texts bounding rectangles is too large. Recall has been defined as the ratio of the number of correct estimates to the total number of targets.

Methods have low recall score when the locating text is not accurate as human marked locations. Following this, the area of intersection divided by the area of the minimum-bounding box containing both rectangles is used, which is termed as an "*m*", and which is the match between the two rectangles. When an exact intersection exists between the two rectangles, the value of *mp* is 0, while in the case of no intersection, the value is 1.

Hence, the best match m(r;R) for a rectangle *r* in a set of rectangles *R* is defined as:

$$m(r;R) = \max\{m_p(r,r') | r' \in R\} \qquad (2.4)$$

Recall is denoted as True Positive rate:

$$Recall = \frac{\sum_{r_t \in E} m(r_t.T)}{T} = \frac{tp}{(tp + fn)} \qquad (2.6)$$

tp means the number of true positives (the number of cases that are positive and classified as positive); *tn* means the number of true negatives (the number of cases that are negative and classified as negative); *fp* means the number of false positives (the number of cases that are negative and classified as positive); and *fn* means the number of false negatives (the number of cases that are positive and classified as negative) (see Figure 2.32).

Then, precision is defined as:

$$Precision = \frac{\sum_{r_e \in E} m(r_e.T)}{E} = \frac{fp}{(fn+tn)} \qquad (2.5)$$

These two measures are combined into a single quality measure f (Lucas et al. 2003)(Lucas et al. 2005):

$$f = 2 \times \frac{(\text{precision} \times \text{recall})}{(\text{precision} + \text{recall})} \qquad (2.6)$$

Furthermore, a Receiver Operating Characteristic curve, or ROC curve, is a graphical plot which represents the ability of the binary classifier system to the diagnostic. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at different threshold settings. The ROC curve is formed by plotting the true positive rate (TPR) as the Y-axis and the false positive rate (FPR) as the X-axis at various threshold settings; the ROC curve illustrates the trade-off between sensitivity and specificity (the increase in sensitivity leads to a decrease in specificity). The area under the curve can be used to measure the accuracy of an experiment; an area of 1 represents a perfect test; an area of 0.5 represents a worthless test.

Text Detection

The dataset for text detection and recognition provides targets with the images which are the ground truth locations for text that is available in the images. It is a form of a rectangle that is bound to a text in the image (Lucas et al., 2003)(Lucas et al. 2005)(Mosleh et al., 2012).

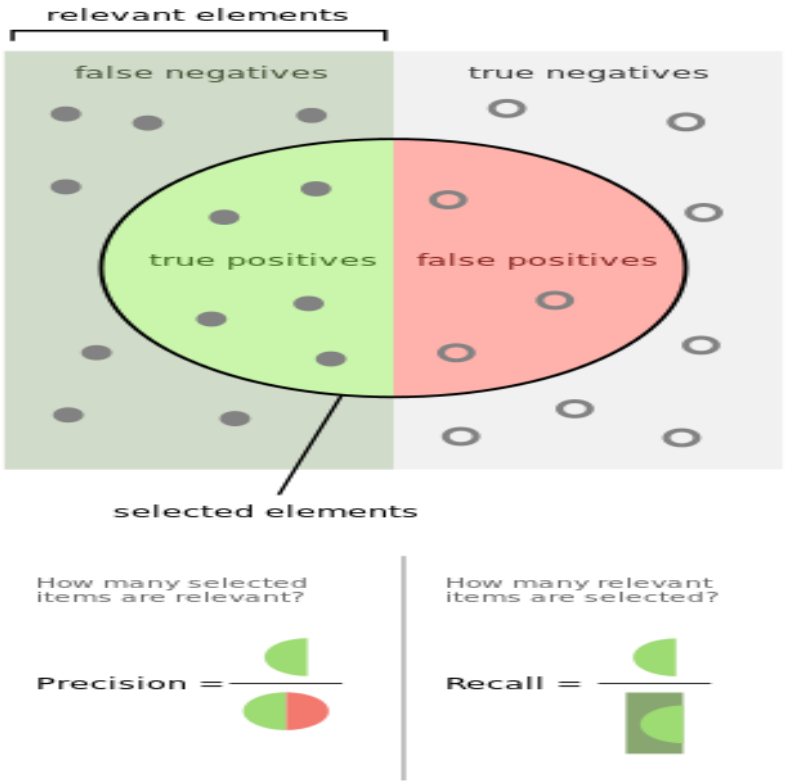


Figure 2.32: Explanation of the precision and recall.

For object detection, the intersection over union (IoU) metric was used to evaluate the accuracy of an object detector in a particular dataset.

There are mainly two different protocols for text detection, namely the IoU-based PASCAL Eval and the overlap-based DetEval. They differ in the criterion of matching predicted text instances and ground truth instances Figure 2.33.

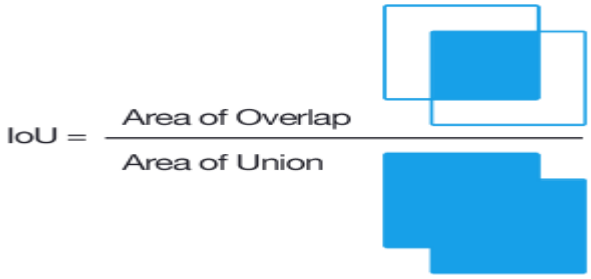


Figure 2.33: Intersection over Union

- PASCAL: The basic idea is that, if the IoU value, i.e. SI / SU, is larger than a designated threshold, the predicted and ground truth boxes are matched together. Pascal protocol uses IoU threshold = 0.5 to specify the true or false positive.

- DetEval: DetEval imposes constraints on both precisions, i.e. SI / SP and recall, i.e. SI/SGT. Only when both are larger than their respective thresholds are they matched together. Threshold = 0.8 is used to specify the true or false positive (Wolf & Jolion, 2006).

Where SGT is the area of the ground truth bounding box, SP is the area of the predicted bounding box, SI is the area of the intersection of the predicted and ground truth bounding box, and SU is the area of the union (Long, He & Ya, 2018).

Most datasets follow either of the two evaluation protocols, but with small modifications.

- **Text Recognition:**

The predicted text string is compared to the ground truth directly. The performance evaluation is in either the character-level recognition rate (i.e. how many characters are recognised) or the word-level (whether the predicted word is 100% correct) (Liu, Meng & Pan, 2019).

- **End-to-End System:**

The evaluation strategy considers both the efficiency of text detection and the capacity of text recognition via precision, recall rate and F-measure(Long, He & Ya, 2018).

2.5 State of the Art

The performances of some state-of-the-art algorithms are provided in the following two sections.

1- Performance of Machine Learning Approaches

The performances of certain techniques encompassing the ICDAR competition winners and other datasets are documented in Table 2.2. The stroke width transform is viewed as a significant language independent technique, and the ‘stroke’ is viewed as a more practical feature for text detecting, especially for the detecting of high-resolution text.

The *P*, *R*, and *F* appearing in tables indicate precision, recall and *F*-measure, respectively. “50” and “1k” are lexicon sizes, “Full” denotes the combined lexicon of all images in the benchmarks, and “None” means lexicon-free, while “WRA” stands for word recognition accuracy. It is worth noting that “strong” means the lexicon containing 100 words specific to

each image; “weak” denotes a lexicon with all words in the testing set; and “generic” means a lexicon containing 90,000 words.

Table 2.2: Text Detection Performance Literature

Literature	**Description**	**Dataset**	**P**	**R**	***F***
(Epshtein et al. 2010)	Stroke Width Transform, Hierarchical clustering	ICDAR’03	0.730	0.600	0.660
(Shivakumara, Phan and Tan, 2011)	Fourier-Laplacian filtering, skeleton analysis	MSRA-I video text	0.810	0.930	0.870
(Mosleh, Bouguila and Hamza, 2012)	Bandlet based Stroke Width Transform, CCA	ICDAR’03	0.760	0.660	0.710
(Yao *et al.*, 2012)	Stroke Width Transform, CCA	ICDAR’03/ MSRA-II	0.688 0.630	0.660 0.630	0.660 0.660
(Koo and Kim, 2013)	MSERs, Agglomerative clustering controlled by an Adaboost classifier	ICDAR’11 scene text	0.830	0.625	0.713
(Ye and Doermann, 2013)	MSERs, component hypothesis extension	ICDAR’11 scene text	0.892	0.623	0.733
(Ye and Doermann, 2013)	MSERs pruning, single-link clustering algorithm with learned distance parameters	ICDAR’13 scene text ICDAR’13 graphic text	0.885 0.938	0.665 0.824	0.759 0.877

The stroke width transform is regarded as an important language independent approach, and the ’stroke’ is regarded as one of the most effective features when it comes to detecting text, and particularly detecting high-resolution text. Epshtein et al. (2010) developed the classic SWT approach, and the Bandlet-based SWT (Mosleh, Bouguila & Hamza, 2012) approach reports better performance on the ICDAR2003 scene text detection benchmark. With the ICDAR2011 scene text dataset, the MSER-based detection approach with learned CCA models (Bai, Yin & Liu, 2012; Mosleh, Bouguila & Hamza, 2012; Koo & Kim, 2013; Ye & Doermann, 2015a) achieves a solid performance. With the ICDAR2013 dataset, a state-of-the-art performance is achieved by introducing a MSER pruning strategy and hybrid feature-based verification

The approach that uses Fourier-Laplacian filtering, component skeleton analysis and geometry feature-based verification (Shivakumara, Phan & Tan, 2011) reports the best performance with the Tan dataset, which contains multilingual and multi-oriented video text. The approach based on SWT, hierarchical clustering and Random Forest classification of hybrid features (Yao et al., 2012) shows state-of-the-art performance with the MSRA-II dataset, which includes multilingual and multi-oriented scene text.

Table 2.3: Text recognition Performance (WRA= Word Recognition Accuracy)

Method	Description	Dataset	Lexicon	WRA
TH-OCR (Karatzas *et al.*, 2011)	Commercial OCR software	ICDAR'11	-	0.412
(Wang, Babenko and Belongie, 2011)	HOG and Random Ferns based character model, pictorial model optimization with a small lexicon	ICDAR'03 SVT	50/1,156 50	0.760/0.620 0.570
(Mishra, Alahari and C. V. Jawahar, 2012)	Integrating language prior and appearance features using CRF	ICDAR'03 SVT	50 50	0.818 0.732
(Novikova *et al.*, 2012)	Large-lexicon driven recognition, weighted finite-state transducers-based inference	ICDAR'03 ICDAR'11 SVT	1165/90k 90k 50	0.828/0.785 0.667 0.729
(Goel *et al.*, 2013)	Holistic recognition by gradient based features and dynamic matching (K-NN)	ICDAR'03 SVT	50 50	0.897 0.773
(Shi, Wang, Xiao, Zhang, Gao, *et al.*, 2013)	Deformable character models, pairwise language priors, CRF based optimization	ICDAR'03 ICDAR'11 SVT	50/1,156 50/1,189 50	0.874/0.793 0.870/0.829 0.735

With respect to word recognition, the performance is improved by integrating appearance models with language priors, as shown in Table 2.3. Recent approaches further improved the performance by using top-down and bottom-up cues (Mishra, Alahari & Jawahar, 2012), and high-order language priors (Novikova et al., 2012). The deformable model based approach (Shi et al., 2013) shows high performance given a small lexicon. Additionally, the Goel et al. (2013) approach using gradient-based features (HOG) for the whole word images and k-nearest neighbour for classification reports state-of-the-art performance.

With respect to end-to-end recognition, as shown in Table 2.4, the performance of all approaches is very low. However, the highest end-to-end recognition accuracy remains lower than 50%, indicating that it remains an open problem.

Prior to the use of deep learning, text detection and recognition methodologies basically depended on hand-crafted tools to extract low-level image features, which required frequent pre-processing and post-processing steps. Those methods were restricted by the lack of hand-crafted features description and the complexity of implementation. They could not deal with complex circumstances such as blurred images (Long et al., 2018).

Table 2.4: End-to-End Text Recognition Performance

Method	Description	Dataset	Lexicon	WRA
(Wang, Babenko and Belongie, 2011)	HOG and Random Ferns based character model, pictorial model optimization with a small lexicon	ICDAR'03 SVT	1,15650	0.510 0.380
(Neumann and Matas, 2012)	MSER based character localization and OCR based recognition with open vocabulary recognition	ICDAR 11	-	0.372
(Weinman *et al.*, 2014)	CCA based detection, Gaussian mixture model segmentation, word recognition with a Semi-Markov model. open vocabulary recognition	ICDAR 11	244K	0.386

2. Performance of Deep Learning Approaches

The performances of some state-of-the-art approaches using deep learning for text detection and recognition are given in Tables 2.5 and 2.6. With the detection of the ICDAR dataset, the precision of current mainstream methods has exceeded 90%, while recall and F-measure still have room for further improvement. Numerous CNN-based methods are very competitive and have achieved state-of-the-art performance in various benchmarks, e.g., Hu et al. (2017), and Lyn et al. (2018).

Table 2.5 shows that the performances of all detection methods on the horizontal text datasets, such as ICDAR2013, were better than the performance on the MSRA-TD500, which is multi-oriented text dataset, and curved text datasets such as CTW1500. This is due to the complexity of the text included in datasets and the proposed methods.

State-of-the-art for text recognition shows that CNN-based methods proposed by Cheng et al. (2017) and Jaderberg et al. (2016) achieved good performance on SVT and IIIT5K. Shi, Bai and Yao (2017) proposed the CNN+RNN framework, which is quite popular with many previous approaches. This resulted from the integration of convolutional features and recurrent layers into the CRNN. It provides a great deal of robustness to noises and distortions, which is utilised by convolutional layers, and the contextual information in the score, which can be utilised by recurrent layers. Moreover, the CRNN model eliminates the fully connected layers, thus helping to build a more compact and efficient model.

Table 2.5: Detection performance on ICDAR2013, ICDAR2015, CTW1500 and MSRA-TD500 using Deep Learning

Method	**ICDAR2013**			**ICDAR2015**			**CTW1500**			**MSRA-TD500**		
	P	***R***	***F***	***P***	***R***	***F***	***P***	***R***	***F***	***P***	***R***	***F***
EAST (Zhou *et al.*, 2017)	92.64	82..6	87.37	83.57	73.47	78.20	78.7	49.1	60.4	87.28	67.4	76.08
WordSup (Yao *et al.*, 2012)	93.34	87.53	90.34	79.33	77.03	78.16	-	-	-	-	-	-
SegLink (Jaderberg, Vedaldi and Zisserman, 2014)	87.7	83.0	85.3	73.1	76.8	75.0	42.3	40.0	40.8	86	70	77
He et al. (W. He *et al.*, 2017)	92	80	86	82	80	81	-	-	-	77	70	74
Lyn et al. (Lyu, Yao, *et al.*, 2018)	93.3	79.4	85.8	94.1	70.7	80.7	-	-	-	87.6	76.2	81.5
TextSnak (Long *et al.*, 2018)	-	-	-	84.9	80.4	82.6	67.9	85.3	75.6	84.2	73.9	78.3

Table 2.6: State-of-the-art performance of recognition across a number of datasets

Method	**IIIT5K**			**SVT**		**IC13**	**IC15**	**SVTP**	**CUTE**
	50	**1K**	**0**	**50**	**0**	**0**	**0**	**0**	**0**
(Jaderberg *et al.*, 2016)	97.1	92.7	-	93.2	71.7	81.8	-	-	-
(Lee and Osindero, 2016)	96.8	94.4	78.4	96.3	80.7	90.0	-	-	-
(Shi *et al.*, 2016)	96.2	93.8	81.9	95.5	81.9	88.6	-	71.8	59.2
(Shi, Bai and Yao, 2017)	97.8	95.0	81.2	97.5	82.7	89.6	-	-	-
(Cheng *et al.*, 2017)	99.3	97.5	87.4	97.1	85.9	93.3	70.6	-	-
(Shi *et al.*, 2018)	99.6	98.8	93.4	99.2	93.6	91.8	76.1	78.5	79.5

State-of-the-art for end-to-end systems shows that the modification of the Mask R-CNN framework, which was proposed by Lyu et al. (2018) achieved state-of-the-art performance in ICDAR2013 benchmarks, while FOTS (Liu et al., 2018), which is a combination of the fully convolutional network to predict the detection bounding boxes and the Recurrent Neural Network (RNN) for recognition, achieved state-of-the-art performance in ICDAR2015 benchmarks Table 2.7.

Table 2.7: State-of-the-art performance of End-to-End on ICDAR2015 and ICDAR2013

Method	ICDAR2015			ICDAR2013		
	S	W	G	S	W	G
(Jaderberg *et al.*, 2016)	-	-	-	86.4	-	-
TextBoxs (Liao *et al.*, 2017)	-	-	-	91.6	89.7	83.9
(He *et al.*, 2018)	82	77	-	91	89	86
FOTS(X. Liu *et al.*, 2018)	83.55	79.11	65.33	91.99	90.11	84.77
Masktextspotter (Lyu, Liao, *et al.*, 2018)	79.3	73.0	62.4	92.2	91.1	86.5

The six tables above show that the performance of deep learning methods is better than the performance of the machine learning methods in the three tasks of text detection, text recognition, and end-to-end.

2.6 Summary

Text, as a vital tool for communication and collaboration, has been playing a more important role than ever in modern society, and therefore text detection and recognition in natural scenes have become important and active research topics in computer vision and document analysis.

Many techniques have been developed to detect and recognise the text in scene images and video. With respect to text detection and localisation, the methods can be roughly classified into the following categories: connected component (CC)-based methods, texture-based methods, and deep learning-based methods. Among them, the most representative approaches of the CC-based methods are MSER (Neumann & Matas, 2011) and SWT (Epshtein, Ofek & Wexler, 2010), which are the basis of various state-of-the-art methods. With the rapid development of deep learning, a large number of approaches (Shi, Bai & Belongie, 2017)(Liao

et al., 2017)(Liu & Jin, 2017)(Cheng et al., 2017)(Bušta, Neumann & Matas, 2017) now adopt neural networks and achieve competitive performance. In addition, multi-oriented text detection has attracted more interest, because it is more challenging and practical. As for scene text recognition, the CNN+RNN framework (Shi, Bai & Yao, 2017) is quite popular with many advances on previous approaches. Moreover, numerous approaches (He et al., 2016)(Lee & Osindero, 2016) take the problem of text recognition as a sequence recognition task. Compared with text detection or recognition, the end-to-end system is more challenging but has direct practical value.

Despite the fact that the machine learning-based methods have achieved competitive results in some challenging situations, it is not easy to extract features which are more abstract than those obtained via deep learning. Reading scene text has greatly benefitted from deep learning-based approaches.

Chapter 3

Feature Descriptors

3.1 Introduction

This chapter presents the theoretical and conceptual background of the work presented in chapter 4 of this thesis. It presents several effective feature descriptors, including the Gray Level Co-occurrence Matrix (GLCM), Histogram of Oriented Gradients, Local Binary Patterns (LBP), as well as classifiers Support Vector Machine (SVM) and Random Forest (RF) classifiers, and Correlation based Feature Selection (CFS).

3.2 Gray Level Co-occurrence Matrix (GLCM)

GLCM is a matrix that plots the angular relationship between adjacent pixels in an image by calculating the frequency of the occurrence of a pixel intensity value with another pixel value (Haralick et al., 1973; Song and Tang, 1997). The matrix is a second-order function that tabulates the frequency of different combinations of pixel brightness values (grey levels) in an image. Common angles of GLCM are shown in Table 3.1 (given the distance D), and the GLCM probability measure (Clausi, 2001) is shown in Equation 3.1.

Table 3.1: GLCM common angles

Angle	Offset
0	[0 D]
45	[-D D]
90	[-D 0]
135	[-D -D]

$$p_{ij} = \{G_{ij} | (\delta, \theta)\}; G_{ij} = \frac{P_{ij}}{\sum_{i,j=1}^{G} P_{ij}} \qquad (3.1)$$

Where P_{ij} is the number of occurrences of gray levels i and j within the given window, given a certain (δ,θ) pair, pixel distance (δ) and orientation (θ); while G is the quantised number of gray levels. Some pixel values and their GLCM representations are illustrated in Figure 3.1.

GLCM considers the relationship between two pixels; the first one is the reference pixel and the other is its neighbour. The number of combinations that occur is calculated where the

result filled a cell is GLCM (Voisin et al., 2013), so as to determine how often a neighbour pixel or gray level relationship (0,0) occurs at the right of a different gray level 0 pixel, or reference pixel, is calculated from the image by the number of times 0,0 is shown for top left cells, Where in respect to the direction, distance between the neighbour and the pixel of interest defines the offset. Figure 3.2 shows the common angles when D is 1 (Haralick et al., 1973; Song and Tang, 1997).

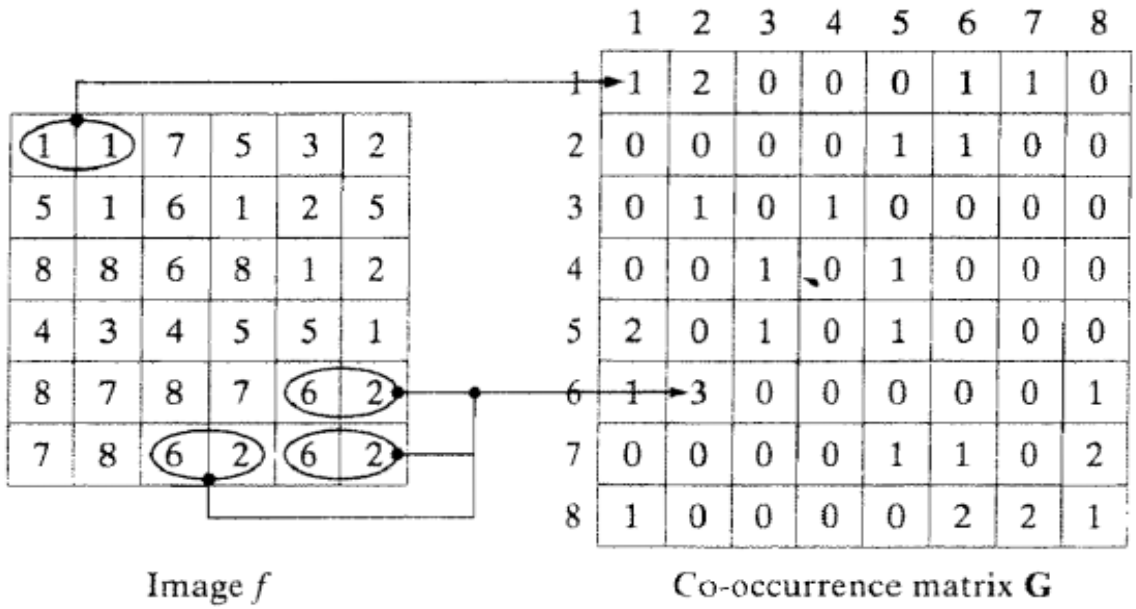


Figure 3.1: Pixel values with the GLCM representation

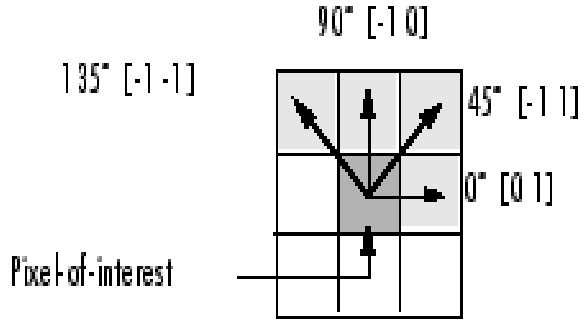


Figure 3.2: Angle and Distance between pixel

Texture features are calculated using GLCM are contrast, correlation, homogeneity, energy and their equations are shown from Equation (3.2) – (3.7) respectively.

3.2.1 Contrast (CON)

It is the sum of squares variance, is another term that describes CON or contrast, and is expressed as:

$$\sum_{i,j=0}^{N-1} P(i,j) \times (i-j)^2 \qquad (3.2)$$

It is the result of the measure of the intensity contrast between a pixel and its neighbor over the whole image. It is reflecting the sharpness and depth of texture furrows. The deeper texture furrows have greater CON values and will be easier to recognise. Conversely, shallow-textured furrows will result in lower CON values. (Haralick, Shanmugam and others, 1973; Clausi, 2002).

3.2.2 Homogeneity (HOM)

Homogeneity is the inverse difference moment, describe the nearness of the distribution of elements in the GLCM to the GLCM diagonal. It is expressed as:

$$\sum_{i,j=0}^{N-1} P(i,j) \times \frac{1}{1+(i-j)^2} \qquad (3.3)$$

Overall, when there is no variance of texture between regions, the value of HOM is high, so that local changes in the image are measured by the inverse difference moment, and the image homogeneity is also reflected. (Haralick, Shanmugam and others, 1973; Clausi, 2002).

3.2.3 Energy and Angular Second Moment (ASM)

It is a value of the summation of squared elements in the GLCM. where the range is = [0 1]. For a constant image the Energy is 1. Energy property is also known by uniformity of energy, and angular second moment, or uniformity equation (3.4). Energy describe the thickness and uniformity of the texture. The value of Energy will be small if GLCM all elements has the same value, In contrast, Energy value will be large when some GLCM elements have greater values than others (Haralick, Shanmugam and others, 1973; Song and Tang, 1997; Clausi, 2002).

$$\sum_{i,j=0}^{N-1} P_{ij}^2 \qquad (3.4)$$

3.2.4 Entropy (ENT)

The information amount contained within an image is defined as ENT or entropy, but information of images is also associated with texture, and it is a random measure. ENT is expressed as:

$$\sum_{i,j=0}^{N-1} P_{i,j}\left(-InP_{i,j}\right) \qquad (3.5)$$

Therefore, the complexity of an image or the non-uniformity of an image is indicated by entropy, so that 1 is the maximum value of entropy, which is achieved when the elements of GLCM have maximum randomness, scattered distribution and all probabilities are equal.

3.2.5 Correlation

There are descriptive statistics of GLCM texture measures; which are derived from GLC matrix. Those descriptors include GLCM mean, GLCM variance. They have a greater consideration of the GLCM matrix, rather than the orderliness and contrast. Pixel values are weighted by occurrence frequency in combination with specific neighbour pixel values are not weighted by a familiar mean equation or regular mean equation, such as the frequency of its occurrence by itself.

Therefore, the gray level linear dependency of neighbouring pixels is measured by GLCM correlation, and shown in the equation below:

$$\sum_{i,j=0}^{N-1} P_{i,j} \left[\frac{(i-\mu_i)((i-\mu_i))}{\sqrt{(\sigma_i^2)(\sigma_j^2)}} \right] \qquad (3.6)$$

where

$$\mu_i = \sum_{i,j=0}^{N-1} i(P_{i,j}), \mu_j = \sum_{i,j=0}^{N-1} j(P_{i,j}), \qquad \sigma_i^2 = \sum_{i,j=0}^{N-1} P_{i,j}(i-\mu_i)^2,$$

$$\sigma_j^2 = \sum_{i,j=0}^{N-1} P_{i,j}(j-\mu_j)^2 \qquad (3.7)$$

GLCM correlation provides different information from other texture measures, and is independent of these, because the calculation is different, and in combination with another texture measure, this is often used. In this case, 1 is correlated perfectly and 0 is not correlated, so that the relationship with actual calculated values is more intuitive, but the variance of GLCM is 0 if images have one gray value. Therefore, when a specific number is divided by 0, the correlation result will be not a number or NaN. Correlation of specific areas of the image is reflected in the correlation value within the GLCM matrix from column elements or row elements that are shown to be similar from the correlation measure of the equation. Therefore, there is a large correlation value if element values of the matrix are equal, but there is a small correlation value if element values of the matrix differ. When images have an overall horizontal

texture, when compared with the COR value of other orientations, there is a larger COR value in this direction (Haralick, Shanmugam and others, 1973).

3.3 Histogram of Oriented Gradients (HOG)

The histogram of oriented gradients (HOG) is a visual descriptor that was originally proposed for the task of human detection (Dalal and Triggs, 2005). This technique identifies the shape and appearance of objects in an image by analysing the distribution of edge directions or intensity gradients. This descriptor is implemented by dividing an image into cells, after which a histogram of edge orientation or gradient direction is compiled for all of the pixels in each cell. The HOG is a combination of the histograms of all cells in the image. This descriptor can be improved by normalising the contrast of local histograms. This is performed by calculating the intensity across the region of the block and using the result to normalise all the cells within the block where the overlapping blocks composed of neighboring cells. This contrast-normalisation leads to improved invariance when there is shadowing or changes in illumination Figure: 3. 3.

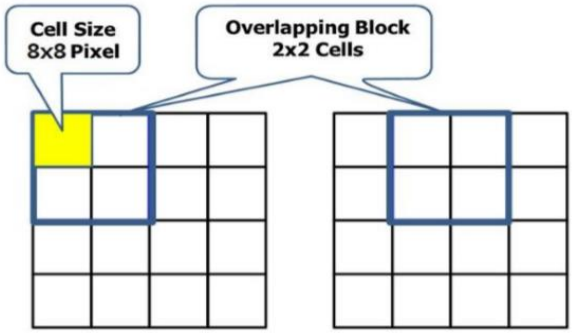


Figure 3.3: Cells and Overlapping Blocks

Equation 3.8 shows that the edges are detected by convolving the image patch both horizontally and vertically. This method requires gray scale images to be filtered with filter kernels.

$$D_X = [-1\ 0\ 1]\ \ and\ \ D_y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \qquad (3.8)$$

The image patch is then subdivided into rectangular cells. Gradients for pixels within each cell were computed. In color images, the largest gradient is chosen after the gradient for each pixel within each channel is separately computed. Given an image I, the derivatives x and y can be obtained using a convolution operation:

$$I_x = I \times D_x \; and \; I_y = \mathrm{I} \times D_y \qquad (3.9)$$

The magnitude of the gradient is

$$|G| = \sqrt{I_x^2 + I_y^2} \qquad (3.10)$$

The orientation of the gradient is given by:

$$\emptyset = \arctan \frac{I_y}{I_x} \qquad (3.11)$$

Within each cell, a weighted vote for cell orientation is computed for each pixel. This vote is weighted by the gradient magnitude, i.e. the L2 norm. The votes of each pixel are collected into orientation bins. Depending on whether there is a signed or unsigned gradient, each vote is placed into the closest bin in the range of 0 to 180 degrees, or 0 to 360 degrees. The gradient is unsigned in this algorithm; therefore, the range is 0 to 180 degrees. These gradients are then stored in a histogram (i.e. the HOG). Dalal and Triggs report that the performance of the algorithm can be enhanced by using a conjunction with nine channels in a HOG for unsigned gradients (Dalal and Triggs, 2005).

As discussed previously, contrast normalisation allows for the suppression of the effect of illumination and contrast changes on the gradient magnitude. This penultimate step is necessary for improved performance. It is achieved by normalising large blocks that contain cells that have been grouped. This ensures that low-contrast regions are stretched. Consistency across the image patch can further be ensured by using overlapping blocks (which considers local variation). Thus, the HOG is a vector of the different components of the normalised cell histograms of all the blocks. Different methods exist for block normalisation.

If v is the non-normalised vector containing all the histograms of a certain block, and $\| vk \|$ is its k-norm (for k=1,2), and e is the constant, then L2-norm (normalisation factor(*f*)) can be calculated using Equation 3.12.

$$\mathrm{L2-norm} \; f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}} \qquad (3.12)$$

The normalised orientation histograms for each cell are compiled and computed in a b × cx× cy–dimensional feature vector, where the number of orientation bins is cx× cy (the number of image cells).

The HOG is a flexible descriptor that can be refined for various different applications. This means that there is a wide variety of choices that need to be made when applying the descriptor, such a cell size, block size, and the number of orientation bins (Dalal and Triggs, 2005).

3.4 Local Binary Patterns (LBP)

Pixels of an image are labelled by the LBP operator or local binary pattern operator, which uses decimal numbers for encoding the local structure surrounding the pixels within an image. Equation 3.13 shows that the eight neighbours of each pixel defined as g_1 up to g_2 are compared by subtraction of the centre pixel value. If the results are positive this is encoded as 1 and if the results are negative this is encoded as 0. Equation 3.14 shows LBPs or local binary values are concatenated from a binary number of each given pixel in a clockwise direction that begins from a pixel of the top-left neighbour. The given pixel is then labelled based on the generated binary number corresponding decimal value (Huang et al., 2011).

Local binary patterns could be expressed as:

Pixel neighbourhood:

$$\begin{pmatrix} g_8 & g_1 & g_2 \\ g_6 & g_c & g_3 \\ g_6 & g_5 & g_4 \end{pmatrix} \tag{3.13}$$

Thresholding:

$$\begin{pmatrix} s(g_8 - g_c) & s(g_1 - g_c) & s(g_2 - g_c) \\ s(g_7 - g_c) & & s(g_3 - g_c) \\ s(g_6 - g_c) & s(g_5 - g_c) & s(g_4 - g_c) \end{pmatrix} \quad s(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases}$$

LBP for pixel:

$$LBP = \sum_{p=0}^{P-1} s(g_p - g_c) 2^P \tag{3.14}$$

3.5 Support Vector Machines (SVMs)

SVMs can be used in the training of classifiers, regresses, and probability densities. It is a renowned technique in statistical learning theory (Shalev-Shwartz and Ben-David, 2013). SVMs can also be used for tasks that involve binary (two-class) classification. They perform pattern recognition for binary problems by determining which separating hyperplane has the maximum distance to the closest points of the training set.

This technique achieves an optimal classification of a binary problem by maximising the width of the margin between the two classes of the problem. This margin is defined as the distance between the discrimination hyper-surface in a n-dimensional feature space and the closest training patterns (support vectors) as shown in the Figure 3.4.

Where the support vectors are the data points that are on the boundary of the slab and closest to the separating hyperplane., with + indicating data points of type 1, and – indicating data points of type –1.

The equation of a hyperplane is expressed as

$$f(x)=x'\beta+b=0 \qquad (3.15)$$

xj is a vector consist of a set of points which are representing the training data. Where y_j is the categories of corresponding data set, yj are either 1 or −1,

b is a real number and β ∈ Rd

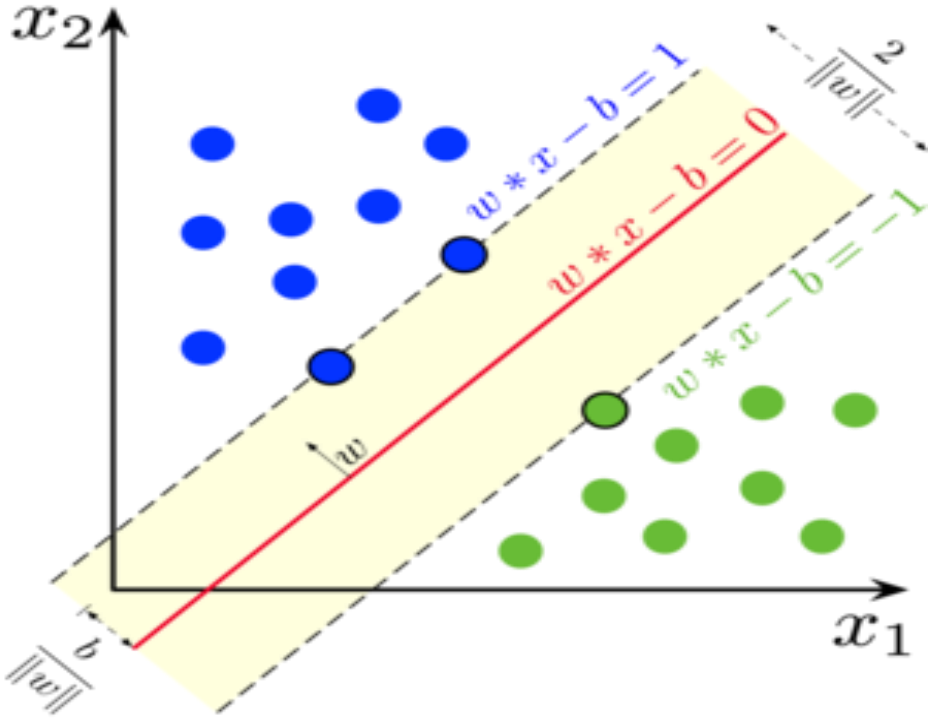


Figure 3.4: Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors

To find the decision boundary which is the best separating hyperplane find β and b that minimize ||β|| such that for all data points (x_j,y_j), $y_jf(x_j) \geq 1$.

The support vectors are the xj on the boundary, those for which $yjf(xj)=1$.

For mathematical convenience, the problem is usually given as the equivalent problem of minimizing $||\beta||$. This is a quadratic programming problem. The optimal solution $(\hat{\beta}, \hat{b})$ enables classification of a vector z as follows:

$$class(z) = sign(\acute{z}\,\hat{\beta} + \hat{b}) = sign\left(\hat{f}\,(x)\right) \qquad (3.16)$$

$\hat{f}\,(x)$ is the classification score and represents the distance z from the decision boundary. A non-linear transformation (Φ) can be applied if the data is not linearly separable in the input space. This maps the data points x ∈ R into a feature space, which is a high-dimensional space (H). This data is then separate (Burges, 1998). Although the original SVM classifier was designed to linearly separate two classes, the problem of separating more than two classes resulted in the development of a multi-class SVM (Nakajima et al., 2003).

3.6 Random Forests

Random forests are classifiers that construct decision trees by some form of randomisation. These classifiers are able to process huge amounts of data at high training speeds based on decision trees. Each tree is binary and created in a top-down fashion Figure 3.5.

The random forest starts the training procedure by selecting a random subset (I) from local training data (I). At the node (n), the training data (I_n) is iteratively split into left and right subsets (I_l and I_r) using the threshold (t) and split function ($f(v_i)$) for the feature vector (v), seen in Equation (3.17) .The threshold (t) is randomly chosen by the split function ($f(v_i)$) in the range

$$\begin{aligned} &t \in (\text{mini } f(v_i), \text{maxi } f(v_i)). \\ &I_l = \{\, I \in I_n \mid f(v_i) < t\}, \\ &I_r = I_n \setminus I_l \end{aligned} \qquad (3.17)$$

Several candidates are then randomly created using the split function as well as the threshold at the split node. The candidate that maximises information gain (ΔE) on its corresponding gain is selected among the candidates above. ΔE is calculated by entropy estimation (Equation 3.18).

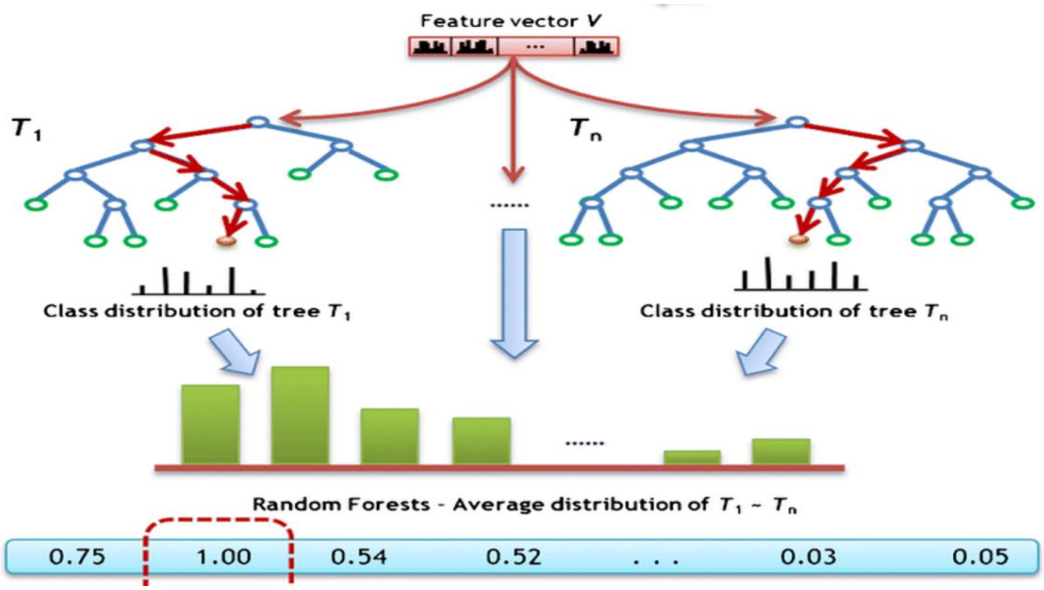


Figure 3.5: Classification process using random forests. In this example, the test image is classified into the second class because it has a maximum posterior probability of 1.0

$$\Delta \mathrm{E} = \frac{|\mathrm{I}_I|}{|I_n|} E(\mathrm{I}_I) - \frac{|\mathrm{I}_I|}{|I_n|} E(\mathrm{I}_I) \tag{3.18}$$

Equation 3.17 shows the Shannon entropy ($E(I)$) of the classes in the set of training images (I). The iterative training process can be ended under two conditions. The first is if no further information gain is possible. The second condition is if a leaf node that is the maximum depth of the tree is reached. Class distributions ($p(c|n)$) of the leaf node are estimated as a histogram of class labels (c_i) of the training examples (i) that have reached the node (n).

The input for the trained random forest is the test image. The final class distribution is generated using an ensemble, or arithmetic averaging, of the distribution of all trees ($L = (l1, l2, \ldots, lr)$) using Equation 3.19.

Equation (3.18) shows the number of trees (T), with the final class of an input image (c_i) if $p(c_i|L)$ has the maximum value (Ko et al., 2011).

$$P(c_i|L) = \frac{1}{T}\sum_{t=0}^{T} P(c_i|l_t) \tag{3.19}$$

Training can be enhanced, and overfitting can be reduced by using an ensemble of tree distributions that are trained only on a small number of random subsets. Although random forests produce a limiting value of the generalisation error, they do not overfit the more trees are added (Bosch, Zisserman and Mu, 2007) (Breiman, 2001). Overfitting is not a problem as these trees always converge according to the strong law of large numbers. Random forests have two important parameters: the depth and number of trees. Increasing the depth of the tree has

been suggested to enhance its performance, although this also increases the memory required for the storage of these trees during experiments (Breiman, 2001).

3.7 Correlation-based Feature Selection (CFS)

CFS is a filtering algorithm that selects subsets of relevant features based on the prediction of the class labels' individual features. This process is defined by (Lu et al., 2014) as follows:

$$Merits_k = \frac{k_{\overline{r_{cf}}}}{\sqrt{k + k(k-1)_{\overline{r_{ff}}}}} \tag{3.20}$$

S_k: the number of features selected in the current subset;

$\overline{r_{cf}}$: average value of all feature-classification correlations;

$\overline{r_{ff}}$: average value of all feature-feature correlations.

The process starts with an empty set of features, but the feature that holds the best discriminative value is added one at a time (Yu and Liu, 2003).

The CFS criterion is defined as follows:

$$CFS = max_{s_k}\left[\frac{r_{cf_1} + r_{cf_2} + \cdots + r_{cf_k}}{\sqrt{k + 2\left(r_{f_1f_2} + r_{f_if_j} + \cdots + r_{f_kf_1}\right)}}\right] \tag{3.21}$$

The r_{cf_i} and $r_{f_if_j}$ variables are referred to as correlations.

CFS involves deciding on two main aspects: the relevance of a feature to the class, and whether this feature is redundant when considering it with other relevant features (Yu and Liu, 2003). A feature is considered significant if it is the main predicting power in a class. The feature selection for classification identifies all of these principal features while removing the remainder of the features. CFS removes the least relevant of two features if they are both found to be redundant. This relevance is linked to the class concept. Thus, CFS retains more information to predict the class (Hall and Smith, 1999).

3.8 Summary

This chapter has provided the reader with the fundamental theoretical and conceptual background of the original and contributor work that is presented in chapters of this thesis.

Where the selected approached are based on texture feature, GLCM represents the overall image texture feature by uses statistical function. It is capable to reflect, the overall average for the degree of correlation between pairs of pixels in different aspects (in terms of homogeneity, uniformity...etc).

LBP has been demonstrated a powerful and computationally simple method to represent local structures and has been extensively exploited in many tasks, such as texture analysis and classification.

Histogram of oriented gradients (HOG) has been proved that is capable to represent the local object appearance and shape within an image. Which can be described by the distribution of intensity gradients or edge directions. The key advantage of HOG descriptor it is run on local cells, it is invariant to geometric and photometric transformations, except for object orientation. Such changes would only appear in larger spatial regions.

More specific theoretical and conceptual knowledge that is required for the individual chapters will be presented within the relevant chapters.

Chapter 4

Text Localization in Natural Images Through Effective Re-Identification of the MSER

4.1 Introduction

The detection and recognition of text in images have a wide variety of applications in information retrieval and document analysis. More importantly, the detection of text in natural scenes is a requirement for further text recognition and image analysis.

The literature on the classification algorithms of text and non-text regions is sparse. Previous research has attempted to use filters to solve the problem of text/non-text region classification. This problem was considered a texture classification problem, which led to the use of different texture descriptors to distinguish between text and non-text regions with the help of machine learning. A number of machine learning techniques have been applied in text detection. These include feature learning (supervised and unsupervised), SVMs, multilayer perceptron (convolutional neural networks), deformable parts models, belief propagation, and conditional random fields.

Hanif S. M., Prevost, L., 2009 extracted three types of features from text segment which are Mean Difference Feature (MDF), Standard Deviation (SD) and Histogram of oriented Gradient (HoG) to create big feature vector, AdaBoost algorithm was used to classify segments to text or non-text (Hanif S. M., Prevost, L., 2009). Anthimopoulose et al, 2010 proposed a modification of Local Binary Pattern (LBP) called edge LBP. The descriptor consists of 256 features extracted from candidate text line by using a sliding window model and Support Vector Machines (SVM) to classify candidate areas (Anthimopoulose et al., 2010). Minelto et al 2011 extended the morphological operation (toggle mapping) to segment urban images. Shape descriptors, Fourier moment, pseudo Zernike moments and polar representation of candidate region used as descriptor and a hierarchical support vector machine as classifier (Minelto et al., 2011). Gonzalez, et al 2012 filtered candidate text regions extracted by MSER by using a set of distinctive features then filtered regions were grouped into lines. Mean Difference Feature (MDF), Standard Deviation (SD) and HOG were used to train SVM with linear kernel to classify lines into text or non_text (Gonzalez, et al., 2012). Zhang et al. 2011 used a mean-shift process to segment candidate text components and then build up a component

adjacency graph. Integrating a first-order components term and a higher-order contextual term, a CRF (Conditional Random Fields) model was used to classify component as text or non-text (Zhang et al., 2011). Trung et al 2012 proposed to use Gradient Vector Flow for the detection of candidate text regions. The detected regions were grouped into text lines by using sizes, positions and colors constraints. HOG and SVM were used to remove false positives using a learning based approach (Trung et al., 2012).

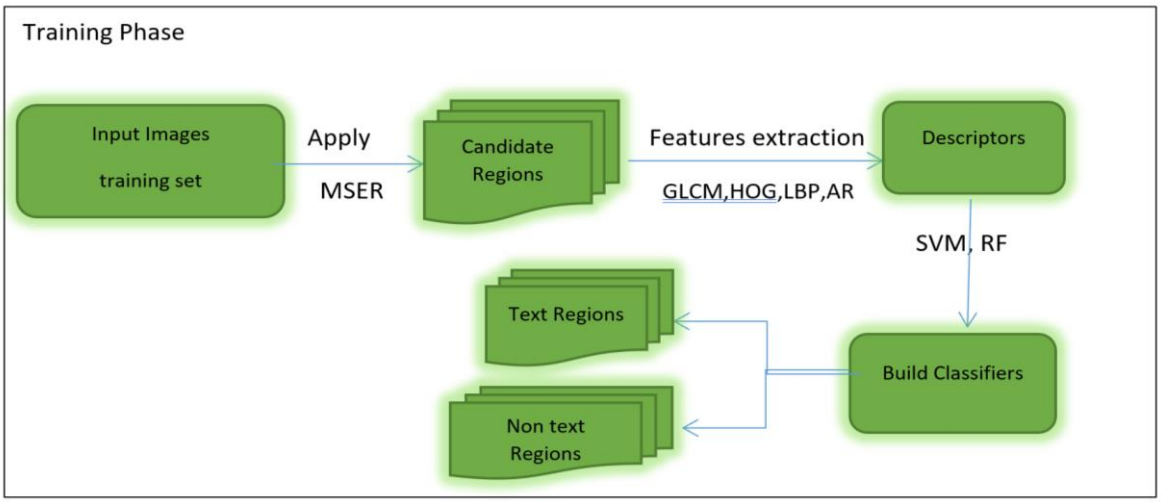
This chapter discusses the design, implementation, and analysis of an accurate method of text detection in natural scene images. MSER is used to detect text-region candidates. This is then followed by a machine learning-based method that is applied to validate and refine the initial detection of text. This chapter will also discuss the efficacy of features that are based on Aspect Ratio, GLSM, LBP, and HOG. The accuracy of detection is maximised using CFS, which selects discriminative features and improves detection results. SVM and RF are also methods used in the classification of text regions.

The advantage of MSER has led researchers to use it for character candidate extraction as it can detect the majority of characters' regions, regardless of their scale, noise, and even illumination variations. However, it can falsely detect non-text regions as text (Matas et al., 2004; Neumann and Matas, 2011). The method proposed here to overcome the problem of detecting non- text regions deals with the classification algorithms of text/non-text regions that are extracted by MSER from the grayscale to obtain text-region candidates. A feature descriptor is calculated using GLCM, LBP, HOG, and the Aspect Ratio. The proposed scheme uses a small set of heterogeneous features which are spatially combined to build a large set of features. The selection and combination of features constitute the two main contributions of the proposed work. All possible combinations between used features were tested in order to obtain the best detection accuracy. By using heterogeneous feature sets in the combination of features, the complexity in the feature selection algorithms is reduced, thus also reducing the overall complexity of the classifier. The computational complexity is an important consideration in real-time applications. Experimental results are illustrated and prove that using a suitable feature selection and combination approach significantly improves the accuracy of the algorithms.

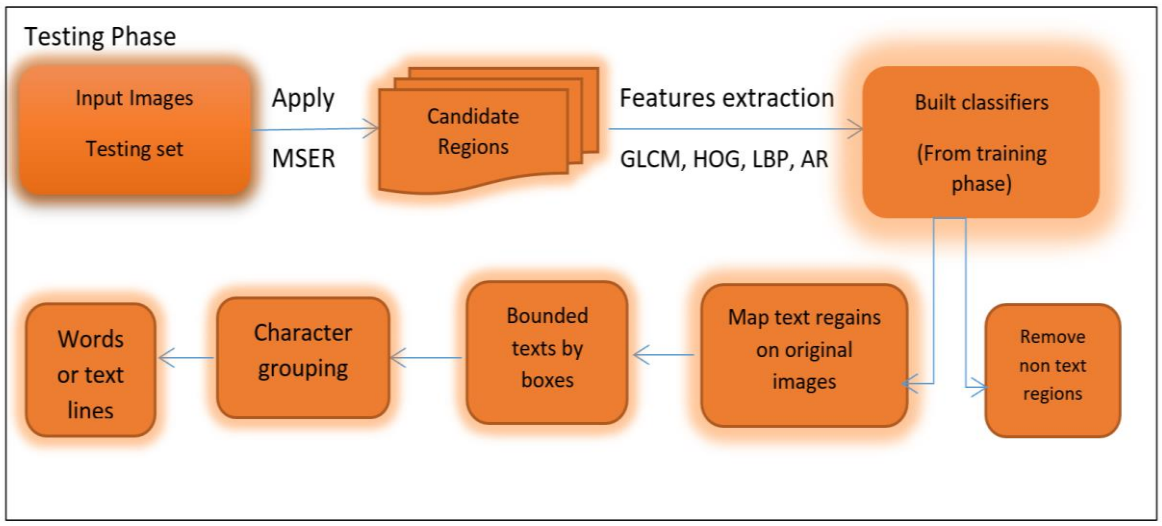
For clarity of presentation, this chapter is divided into three further sections. Section 4.2 presents the proposed method. Section 4.3 puts forth the experimental results and a comprehensive analysis of the performance of the proposed system, before Section 4.5 finally concludes the chapter.

4.2 Proposed Method

An overview of the proposed system is illustrated in Figure 4.1.



(a)



(b)

Figure 4.1: Block Diagram of the text localisation of the proposed method: (a) Training Phase, (b) Testing Phase

MSER regions are used to predict text parts instead of having to create feature descriptors for every pixel/region, which can be computationally expensive. MSER is used to obtain text-region candidates from the grayscale image. MSER detection delivers a list of possible text regions, following which machine learning-based classifiers are employed to refine the detected regions. For each MSER region, image features are calculated using GLCM, LBP, HOG and the Aspect Ratio descriptor. Figure 4.1 shows the flowchart of the algorithm. At the training phase, a training set of text and non-text regions was collected from the ICDAR 2003

dataset, and the resulting classification model was saved for the testing phase. In the testing phase all MSER regions were extracted from each image in the testing set. Candidate regions were classified by using the classifiers built in the training phase based on the descriptors. All MSER regions that are reclassified as text will be mapped back onto the image. Pixels inside these boxes will be marked as text regions.

4.2.1 Training phase

The ICDAR 2003 dataset (Lucas et al., 2003; Lucas, 2005) has been widely used as a benchmark for researchers who work in the field of text detection. There are 509 completely annotated text images included in this dataset. Where 251 of these images are employed for testing and 258 for training. The texts in this database vary greatly in font, size, style and appearance. The dataset provides targets with the images, which are the ground truth locations for text that are available in the images. The targets are used to calculate a precise evaluation of the results of text detection techniques. The text detection methods provide estimates, which take the form of rectangles that are bound to a text area in the image (Lucas et al., 2005; Mosleh, Bouguila and Hamza, 2012). In the experiments conducted in this chapter, 7,423 regions were extracted from the ICDAR 2003 dataset to train and test different descriptors. A total of 6,353 positive patches and 1,070 negative patches were randomly sampled from the training set of the ICDAR 2003 dataset. Figures 4.2 and 4.3 illustrate examples of text and non-text regions respectively.

Figure 4.2: Examples of positive samples

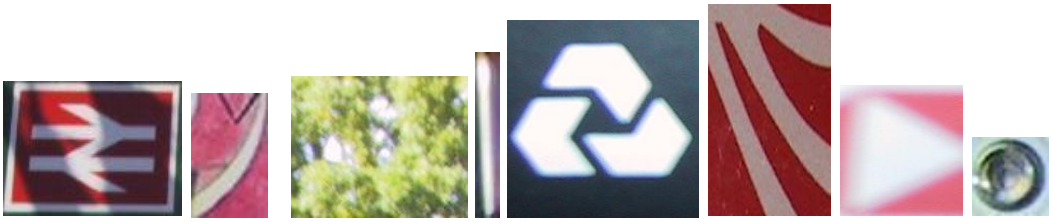


Figure 4.3: Examples of negative samples

Positive samples are represented by the label 1 in the classifier and negative samples are represented by the label 0 in the classifier. Classifiers based on SVM with linear kernel, Multilayer Perceptron (MLP) and Random Forest (RF) were experimented with.

Since the system proposed uses discriminative models for MSER regions labelling, the feature-set was built by using the following features extractor.

1. GLCM Descriptors/Features

Features that are based on the gray level co-occurrence matrix (GLCM) are widely used in images analysis. The GLCM computes the frequency of occurrence of different combinations of gray levels in an image or a section of an image. Textual information is captured by computing the four traditional features based on the GLCM: energy (ASM), contrast (CON), correlation (COR), and homogeneity (HOM). These four features are used to validate the detection of text regions and in the elimination of text-like false positives.

In this implementation, the GLCM has been calculated in four orientations (0º, 45º, 90º, 135º); this is because GLCM is not direction invariant, and texts are located in different directions in the images. To solve this invariant problem, four main directions were defined to detect texts.

Zhuo et al. (Zhuo, Lin and Gu, 2014) proposed the idea that there are relationships between distance and the calculated values of CON, COR, and HOM, etc. GLCM considers the relationship between two pixels; the first one is the reference pixel and the other is its neighbour, distance is the space between those two pixels. As shown by the aforementioned paper, using a Markov Random Field (MRF) could prove that a calculation is correct only when the distance is greater than the value of a GLCM feature. Conversely, when the distance is small, the results of GLCM calculations are random or change (Alsadegh and Lu, 2015). The same idea was proposed by Chaddad et al. and Alsadegh (in Chaddad et al., 2014; Alsadegh and Lu, 2015); in their opinion, when the distance is small, or the two chosen pixels are close together, the result of GLCM calculations rapidly changes with any increase of distance. However, when the distance becomes large, the result will be more stable. The conclusions of these two papers are the same. As a result, it is essential to find a suitable value for the distance. For this reason, a distance of three pixels has been used.

After the four texture features for GLCM detection were selected (energy (ASM), contrast (CON), correlation (COR) and homogeneity (HOM)), the classifiers could be trained. First of all, the mean and variance of correlation, entropy and homogeneity were calculated for a total

of 6,185 texts regions in the dataset. These features were set as the positive samples represented by 1 in the classifier. Following this, the mean and variance of correlation, entropy and homogeneity were also calculated for the non-text region in the dataset. These features were set as the negative samples, represented by 0 in the classifier. For this purpose, we used a classifier based on SVM with linear kernel.

The results show that multilayer perceptron and Random Forest yield almost the same result, but different times compared with SVM, which is give less result and time. Table 4.1 shows that MLP and RF achieved 0.971% for *f* Measure while SVM achieved less accuracy. With regards to the time SVM consumed time less than RF and MLP.

Table 4.1: The results of detection using the GLCM feature and SVM, MLP, RF

Classifier	TP Rate	FP Rate	Precision	Recall	F Measure	ROC	Time/ sec
SVM	0.956	0.055	0.956	0.956	0.956	0.950	0.02
MLP	0.971	0.029	0.971	0.971	0.971	0.992	0.87
RF	0.971	0.029	0.971	0.971	0.971	0.992	0.43

2. Aspect Ratio

Since the Aspect Ratio of most letters of the English alphabet is close to 1, this feature is useful in filtering out false character candidates. In order for elongated letters such as ‘I’ or ‘i’ to be considered, the threshold for their detection should be small enough. The utilisation of the Aspect Ratio of letters can be used for many alphabets and languages. If a letter has a very small Aspect Ratio and is filtered out, the grouping stage will not be affected as the absence of this letter will not affect the grouping of the entire word.

The results show that Random Forest gives the best classification accuracy. Indeed, SVM produces the lowest accuracy and the three classifiers consume almost the same classification time Table 4.2.

Table 4.2: The results of detection using Aspect Ratio and SVM, MLP and RF

Classifier	TP Rate	FP Rate	Precision	Recall	F Measure	ROC	Time/sec
SVM	0.671	0.671	0.450	0.671	0.538	0.500	0.4
MLP	0.698	0.587	0.705	0.705	0.621	0.362	0.45
RF	0.759	0.278	0.767	0.767	0.767	0.813	0.4

3. Local Binary Pattern (LBP) Features

Local Binary Pattern (LBP) is a highly discriminative method for texture segmentation. It is useful for more demanding image analysis, since it is computationally efficient and invariant to monotonic gray level changes. LBP has been shown not only to capture text characteristics, but also local structure characteristics (Zhang et al., 2006). This is useful for text detection. Text is seen by the LBP feature as a composition of strokes similar to the patterns produced by the LBP operator. LBP is therefore efficient in extracting textual features that are inherent within character strokes. Said useful feature was employed in the present study to solve this specific problem.

Pixels can be either more or less intense than the central pixel (Figure 4.4). The image region is considered flat (featureless) when the surrounding pixels are either all black or all white. Uniform patterns, comprising continuous black or white pixels, are interpreted as corners or edges. Non-uniform patterns are defined as the alternation between black and white pixels (Pietikäinen et al., 2013; George and Zwiggelaar, 2019).

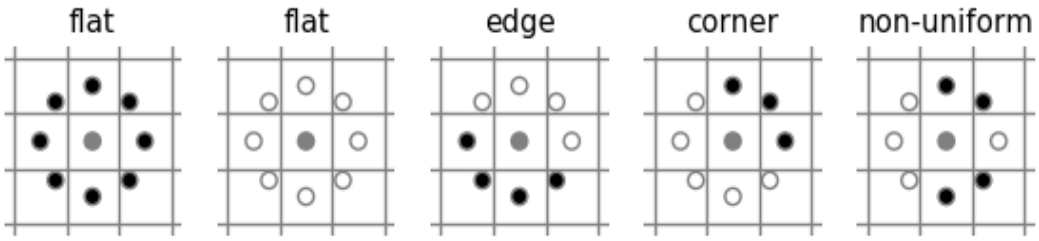


Figure 4.4: Different textures detected by the LBP

LBP uses eight pixels in a 3 x 3 pixel window. This allows the LBP operator to have an unlimited size of adjacent pixels and sampling points. Figure 4.5 illustrates the neighbourhoods used to calculate LBP (Pietikäinen et al., 2013)(George and Zwiggelaar, 2019).

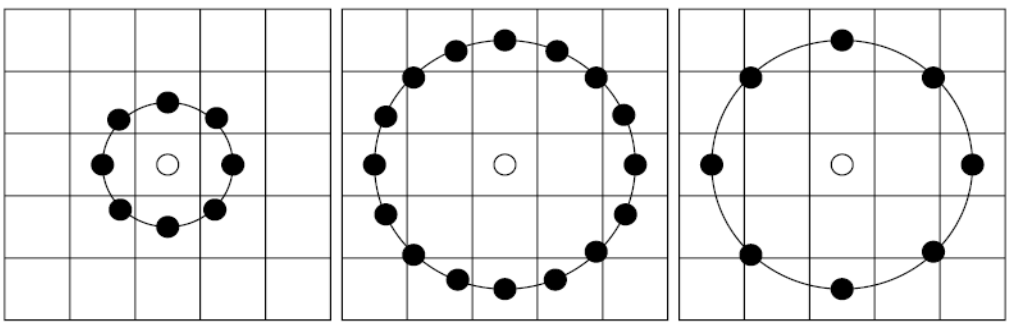


Figure 4.5: Three circular neighbourhood examples (8, 1), (16, 2) and (8, 2) used to define a texture and calculate a local binary pattern (LBP)

Looking at the tables (4.4 to 4.5) above, it can be concluded that a small cell size gives more LBP feature information, which can achieve greater classification model accuracy. It is also noticed that, as the number of LPB descriptors increases, more computational resources are needed, which entails a longer processing time. The three classifiers (SVM, MLP, RF) which were used in the experiments give the best result when the cell size is smaller.

Table 4.3: The results of detection using the LBP feature with 32 cell size

Classifier	TP Rate	FP Rate	Precision	Recall	F Measure	ROC	Time/Sec
SVM	0.820	0.253	0.816	0.820	0.817	0.780	0.15
MLP	0.904	0.104	0.904	0.904	0.904	0.968	24.08
RF	0.912	0.115	0.916	0.912	0.910	0.984	0.65

Table 4.4: The results of detection using the LBP feature with 16 cell size

Classifier	TP Rate	FP Rate	Precision	Recall	F Measure	ROC	Time/Sec
SVM	0.850	0.196	0.850	0.850	0.850	0.826	0.65
MLP	0.893	0.140	0.893	0.893	0.893	0.937	27.92
RF	0.896	0.176	0.897	0.896	0.893	0.951	0.85

Table 4.5: The results of detection using the LBP feature with 8 cell size

Classifier	TP Rate	FP Rate	Precision	Recall	F Measure	ROC	Time/Sec
SVM	0.915	0.088	0.918	0.915	0.915	0.913	1.69
MLP	0.912	0.139	0.912	0.912	0.911	0.961	76.19
RF	0.901	0.167	0.912	0.901	0.897	0.979	1.98

4. HOG Features

The principle behind HOG is that the appearance and shape of local objects are well-characterised by the distribution of local intensity gradients or edge directions (Dalal and Triggs, 2005). For the present study, HOG features were extracted using different cell sizes. These were then used to produce and compute different feature sets of different lengths. Normalization has been performed by grouping small cells into larger blocks with overlap, and each block normalized separately. The final descriptor is the vector of the features calculated from all cells see section 3.3 and Equation (3.12).

The training and testing were categorised into various cell sizes, and for each cell size different feature lengths were extracted (Table 4.6) Figure 4.6.

In each combination of cell size and block size, the number of orientation histogram bins was set to nine, which provided a reasonably low dimensional feature vector that delivered good descriptive power and resulted in better classification accuracy. Because the HOG feature is a texture-based detection method, the samples should be selected from those featured. The sizes of the HOG feature descriptors vary depending on the cell size and block size. These descriptors of both positive and negative samples were applied to the SVM.MLP, RF classifiers to create the classification model, following which the classification performances of each model were evaluated. The most accurate model was used as the HOG parameter setting for the detection system. Tables 4.7, 4.8 and 4.9 show the result of each classifier with different cell size.

Table 4.6: Cell size, block size and features set length

Cell size	Block size	Block overlap	Features length
50 × 50	2 × 2	1× 1	36
32 × 32	2 × 2	1× 1	144
25 × 25	2 × 2	1× 1	324

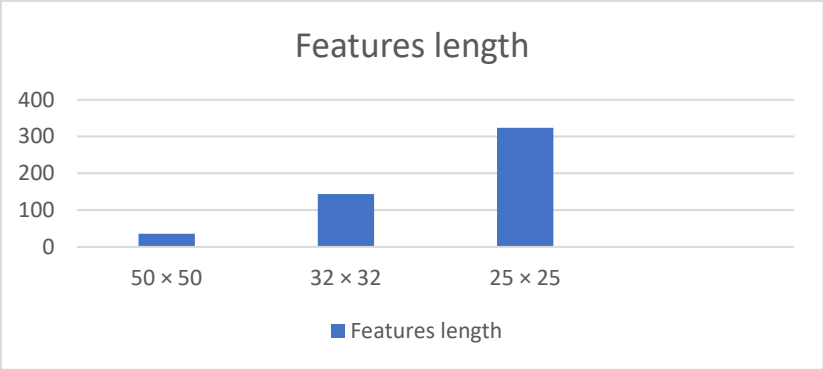


Figure 4.6: The relation between cell size and the number of features

Table 4.7: The results of classification using the HOG with cell size 50×50 feature

Classifier	TP Rate	FP Rate	Precision	Recall	F Measure	ROC	Time/ Sec
SVM	0.784	0.311	0.779	0.784	0.779	0.737	0.14
MLP	0.840	0.158	0.851	0.840	0.843	0.910	7.78
RF	0.870	0.194	0.869	0.870	0.868	0.931	0.64

Table 4.8: The results of classification using the HOG with cell size 32×32 feature

Classifier	TP Rate	FP Rate	Precision	Recall	F Measure	ROC	Time/Sec
SVM	0.825	0.219	0.825	0.825	0.825	0.803	0.65
MLP	0.879	0.140	0.881	0.879	0.880	0.938	97.97
RF	0.898	0.149	0.897	0.898	0.897	0.953	0.92

Table 4.9: The results of classification using the HOG with cell size 25×25 feature

Classifier	TP Rate	FP Rate	Precision	Recall	F Measure	ROC	Time/Sec
SVM	0.856	0.159	0.860	0.856	0.857	0.848	1.49
MLP	0.889	0.120	0.893	0.889	0.890	0.950	601.18
RF	0.902	0.156	0.902	0.902	0.900	0.958	1.96

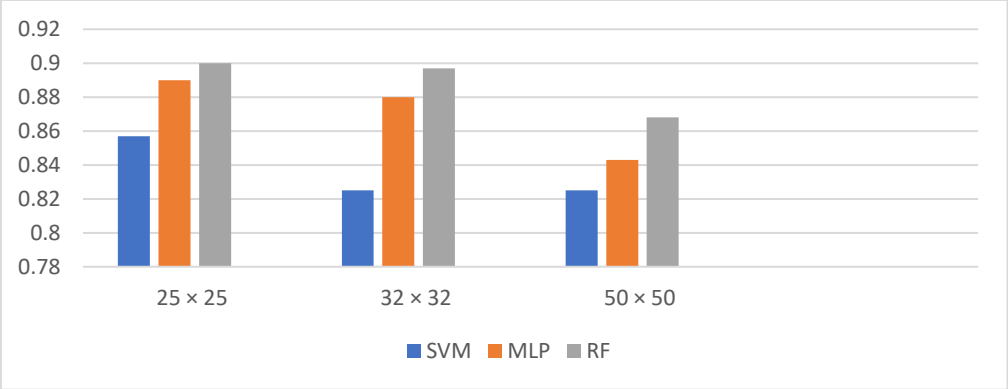


Figure 4.7: F Measure rate for different cell size with different classifier

Looking at the tables (4.7 to 4.9) and Figure 4.7 above, it can be concluded that a small cell size gives more HOG feature information, which can achieve greater classification model accuracy. It is also noticed that, as the number of HOG descriptors increases, more computational resources are needed, which entails a longer processing time.

- **Combination of Multiple Types of Features**

In order to classify text regions, we captured appearance and shape information using the proposed feature sets; in doing so, the GLGM, Aspect Ratio, LBP Histogram and HOG features were extracted from the MSER regions to form a combined feature vector for classification of text candidates. The following combinations of feature sets were studied to determine the possible best feature set: 1) AR and GLCM (AGLCM); 2) AR and LBP (ALBP); 3) AR and HOG (AHOG); 4) GLCM and LBP (GLBP); 5) GLCM and HOG (GHOG); 6) HOG and LBP

(HLPB). LBP, GLCM and AR (LBGLA), HOG, GLCM and AR (HOCLA), HOG, LBP, GLCM and AR (HOLBCLA).

The results from table 4.10 show that the combination of all features gives the best accuracy, while SVM is the best classifier. The combination of LBP+GSLM+AR and the combination of LBP+GSLM give the second-best result (see Figure 4.8).

Table 4.10: The accuracy of using a combination of features

Combination of features	TP	FP	P	R	F	ROC
AR+GLCM (AGLCM)	0.972	0.037	0.972	0.972	0.972	0.964
AR+LBP(ALBP)	0.905	0.105	0.907	0.905	0.906	0.899
AR+HOG (AHOG)	0.912	0.083	0.915	0.912	0.913	0.915
LBP+GLCM (GLBP)	0.987	0.023	0.987	0.987	0.987	0.980
HOG+LBP (HLPB)	0.937	0.069	0.937	0.937	0.937	0.930
HOG+GLCM (GHOG)	0.983	0.021	0.983	0.983	0.983	0.981
LBP+GLCM+AR (LBGLA)	0.987	0.023	0.987	0.987	0.987	0.980
HOG+GLCM+AR (HOCLA)	0.983	0.021	0.983	0.983	0.983	0.981
HOG+LBP+GLCM+AR (HOLBCLA)	0.993	0.012	0.993	0.993	0.993	0.988

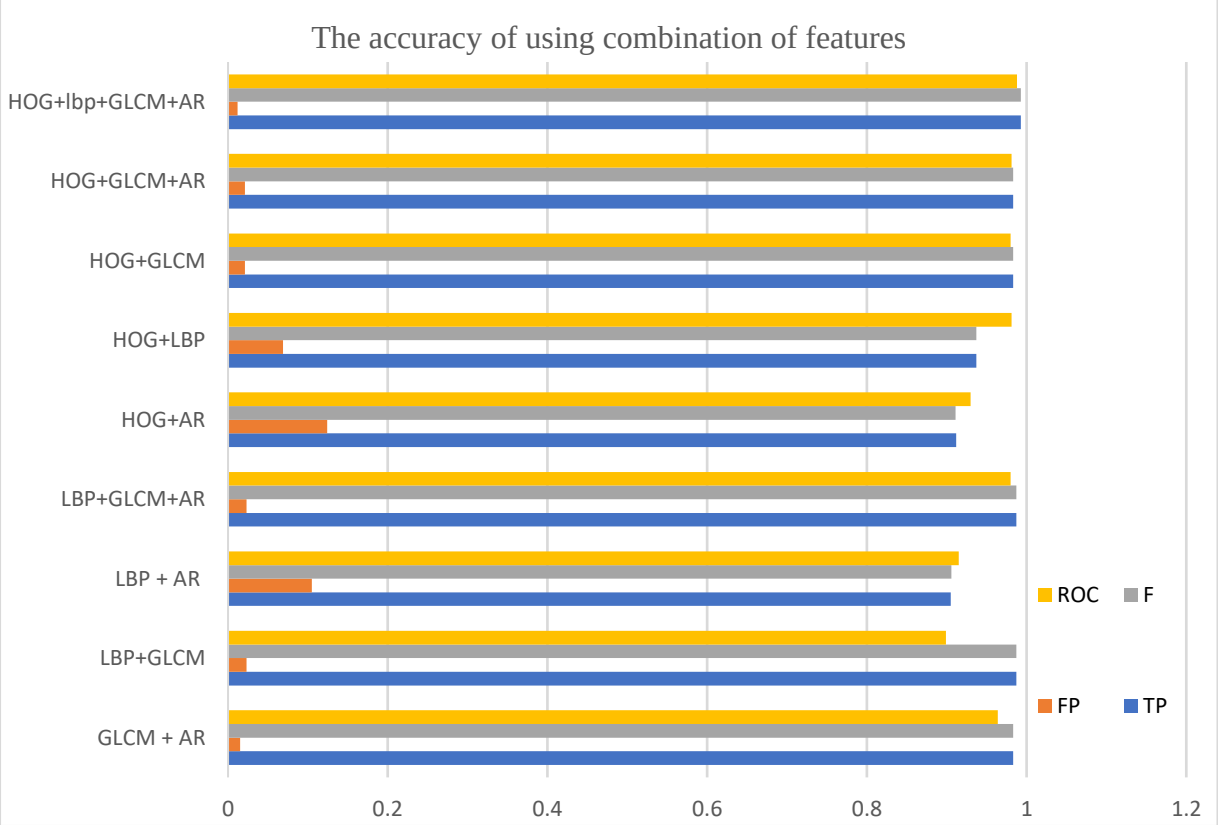


Figure 4.8: The accuracy of combination of features

- **Combination of selected from multiple types of features**

The CFS approach was used to reduce feature space and accelerate the processing cycle. CFS aids in ranking feature subsets according to correlation based on heuristic merit (Hall and Smith, 1999; Lu et al., 2014). This method resulted in reduced numbers of original feature attributes obtained from descriptors of text candidate regions.

Table 4.11: Classification accuracy results with selected features

Selected Features	TP	FP	F	ROC	No. of Selected Features
HOG+LBP	0.925	0.113	0.924	0.972	65+114
LBP+AR	0.923	0.120	0.921	0.986	114+1
HOG+LBP+AR+GLCM	0.949	0.083	0.949	0.996	65+114+1+4
LBP+GLCM	0.943	0.092	0.942	0.992	114+4
LBP+GLCM+AR	0.984	0.027	0.984	0.981	114+4+1
HOG+GLCM+AR	0.956	0.051	0.956	0.954	65+4+1

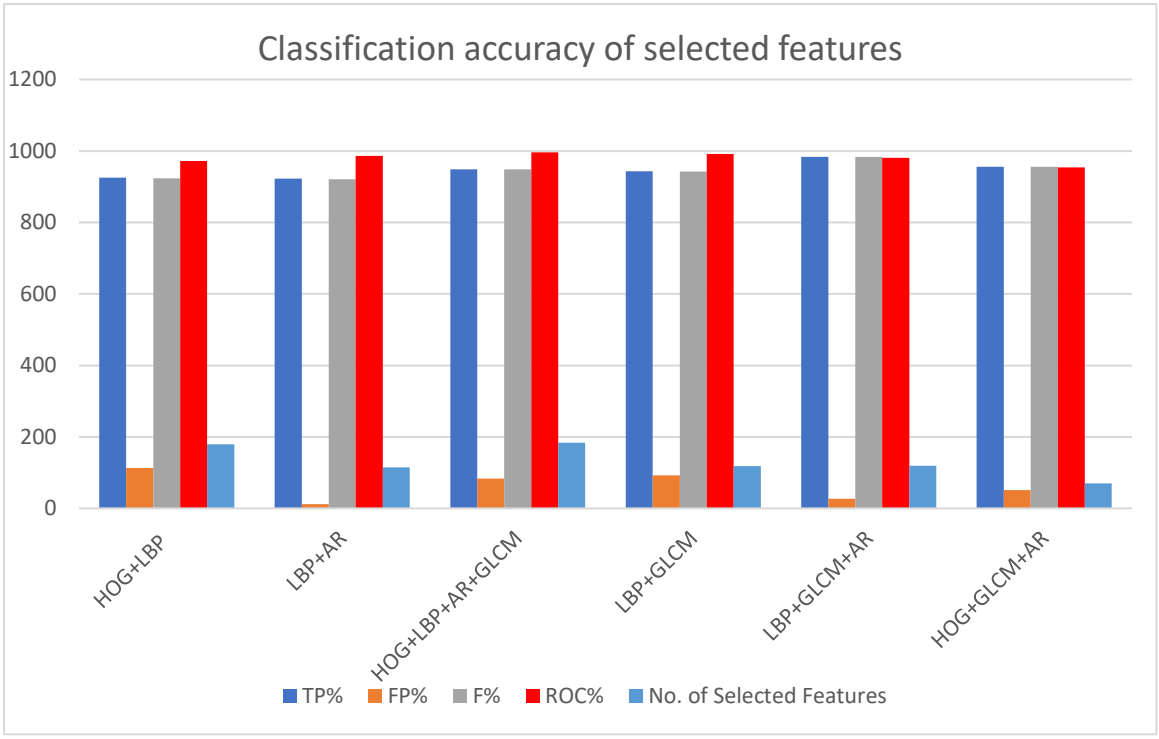


Figure 4.9: Classification accuracy of selected features

Table 4.11 and Figure 4.9 show the classification accuracy of the above six selected features, which are combinations and give good results. However, the combinations of HOG,

GLCM and AR produce the best accuracy with SVM as a classifier. Furthermore, the results show that no relation between the quantity of features and the classification accuracy.

4.2.2 Candidate region re-identification (Test and Evaluate Phase)

The first step in the text detection module is the extraction of MSER in a given image to obtain text-region candidates. The MSER algorithm is dependent only on the intensity of the image. Since text in images is usually of equal intensity, the output of this first step is a list of candidate regions that contain at least one symbol.

The aim of the proposed method is the detection of as many text components as possible, which is difficult, and nearly impossible. Therefore, to recover missed characters, the threshold of MSER is set to the lowest value of 1, which makes it possible to capture text from even the most challenging images.

Figure 4.10 illustrates the results of MSER detection, showing that the algorithm detects many false positives (i.e. non-text regions). The classification model is used to classify regions into either text or non-text. The best descriptors and classifiers – determined in the training phase – are used as a combination of all features: HOG, LBP, GLSM, AR, and RF. All regions classified as non-text by MSER are then removed from the scene image. The remaining text regions are mapped onto the image and bounded by boxes. The pixels inside these boxes are marked as text pixels (Figures 4.10 and 4.11).

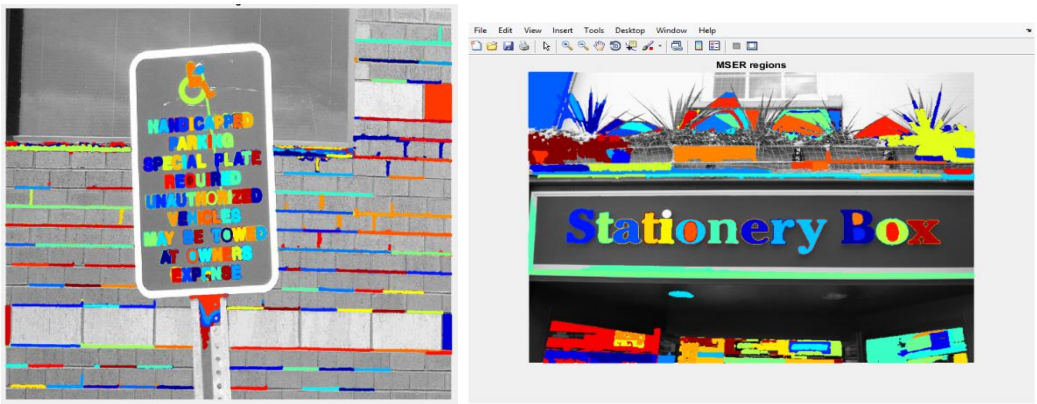


Figure 4.10: The result of using MSER detector

Figure 4.11: Non-text regions are removed using SVM classifier learnt from the combination of features, b: candidate characters bounded by boxes

4.2.3 Text Localization

The character grouping module adjoins characters detected as text into text regions. These regions can either be single words or lines of text. This important step allows the recognition of words in an image and provides more relevant information instead of only individual characters. This step is also important in the optical character recognition (OCR) of words that occur in text regions which are largely connected. The merging of individual text regions into words or text lines can be achieved by finding adjacent text regions, and then forming a bounding box around these regions. Adjacent regions are found by expanding bounding boxes, computed earlier with the region props function, which overlap the bounding boxes of adjacent

text regions. This forms a chain of overlapping bounding boxes of text regions that are part of the same word or text line Figure 4.12.

Figure 4.12: The result of expanding bounding box of text

The overlapping bounding boxes are then merged to form a single box around individual words or text lines. This requires a calculation of the overlapping ratios between neighbouring bounding boxes. Non-zero overlapping ratios would therefore indicate possible adjacent characters in words or in different text lines Figure 4.13.

Figure 4.13: The result of text localisation

4.2.4 Results

The proposed method has been evaluated using several public test datasets and compared against several state-of-the-art text detectors described in the literature. Specifically, the

proposed method compared with the contestants of the ICDAR Challenge (Lucas, 2005), as well as with the detectors of (Epshtein, Ofek and Wexler, 2010)(Chen et al., 2011)(Pan, Hou and Liu, 2011)(Neumann and Matas, 2012)(Yi and Tian, 2012)(Yao et al., 2013). Tables 7 and 8 show the results obtained for each dataset.

Table 4.12: Text detection scores of proposed method and other detectors on the ICDAR 2003 dataset (%)

Algorithms	Precision	Recall	F Measure
Proposed Method	0.85	0.79	0.81
Ye Q. and Doermann, D (Ye and Doermann, 2013)	0.892	0.623	0.733
Yin et al. (Yin, Huang and Hao, 2013)	0.86	0.68	0.76
Neumann and Matas (Neumann and Matas, 2012)	0.85	0.68	0.75
Shi et al. (Shi, Wang, Xiao, Zhang and Gao, 2013)	0.83	0.63	0.72

The training data on the ICDAR 2011 datasets was not applied for testing in experiments. Tables 4.12 and 4.13 show that the proposed method accomplished excellent performance on both datasets and the improvements were significant in terms of Precision, Recall, and F Measure. The increase in terms of the proposed method is mainly due to the combinations of different types of features (HOG+LBP+GLSM+AR).

Table 4.13: Text detection scores of proposed method and other detectors on the ICDAR 2011 dataset (%)

Algorithms	Precision	Recall	F Measure
Proposed Method	0.85	0.83	0.83
Yi, C. and Tian, Y. (Yi and Tian, 2012)	0.73	0.67	0.70
Pan et al. (Pan et al., 2011)	0.67	0.70	0.69
Yao, C. et al. (Yao et al., 2012)	0.69	0.66	0.67
Chen, H. et al. (Chen et al., 2011)	0.73	0.60	0.66
Epshtein et al. (Epshtein et al., 2010)	0.73	0.60	0.66

4.3. Conclusions

In this chapter a text detection and localisation method are presented. The proposed method improved text detection using MSER through a re-identification step using classification models learnt from GLCM, LBP, HOG, Aspect Ratio, combinations and selections of these features. The re-identification performances of SVM, MLP and RF classifiers are compared regarding accuracy. A combination of HOG+LBP+GLCM+AR gave the best accuracy followed by the combination of LBP+GLCM on the tested dataset.

The LBP, HOG and GLCM are based on texture features; GLCM represents the overall image texture feature by using the statistical function. Both HOG and LBP try to utilise the

same kind of information: gradients around a pixel. How HOG and LBP methods use the gradient information is the main difference between them. The robustness of LBP derives from the fact that it uses all eight directions for each pixel, whereas HOG uses one direction for each pixel. However, the roughness of the binning utilised by LBP makes its information loose compared to the information produced by HOG.

HOG is effective at capturing edges and corners in images, while LBP captures the local patterns. HOG and LBP obtain various types of information, which makes them complimentary to each other. HOG and LBP can be combined in image processing applications such as text detection and recognition.

The ICDAR2003,2011 dataset was used as a benchmark in our experiments. After text pixel regions were confirmed, character grouping based on the overlapping ratio of bounding boxes was employed to join pixel regions to word regions or text lines, enabling fast text recognition when using off-the-shelf OCR. In terms of our future work, we aim to improve the feature selection method using deep learning, so as to find more discriminative features and achieve better robustness.

It has been demonstrated that LBP is a powerful and computationally-simple method with which to represent local structures and has been extensively exploited in many tasks, such as texture analysis and classification.

The histogram of oriented gradients (HOG) has been proved to be capable of representing local objects' appearance and shape within an image. This can be described by the distribution of intensity gradients or edge directions. The key advantage of the HOG descriptor is that it is run on local cells, and it is invariant to geometric and photometric transformations, except for object orientation. Such changes would only appear in larger spatial regions.

Chapter 5

Datasets and Data Augmentation

5.1 Introduction

The process of text detection and recognition in natural images remains challenging. That is because of the diversity and complexity of characters presented in natural images, such as appearances varying and distortion from different camera angles, low resolution, illumination, occlusion, mixed within complex background, as well as many different styles and sizes etc. Moreover, texts are often mixed with complex backgrounds which makes the process more difficult and challenging.

The lack of sufficient labelled training datasets has become a main issue that affects the promotion of scene text detection and recognition research. This encourages researchers to build new datasets, as well as develop new methods to generate variations of samples from existing dataset samples using for example Data augmentation technique (Simard, Steinkraus and Platt, 2003). Data augmentation is an approach to increase the training data from the original dataset by applying transformation while keeping the original class of data and labelling. Data augmentation has been approved effective to improve training results in many computer vision tasks such as segmentation of brain electron microscopy images(Valle et al., 2017) detection of melanoma (Valle et al., 2017) and gastrointestinal disease detection from endoscopical images (Asperti and Mastronardo, 2017).

Few data augmentation libraries are available to generate augmented images from original images for example classification such as Augmentor (Bloice, Stocker and Holzinger, 2017), Imgaug (Jung, 2017). Furthermore, deep learning frameworks, such as Keras (Chollet, 2015) and Tensorflow (Abadi et al., 2016) provide augmentation functions. However, many of these augmentation libraries are designed to deal with classification task, while those libraries do not provide effective argumentation tools for object detection and localization tasks. Visual-based detection and localization could take significant advantage of data augmentation but existing augmentation libraries for these tasks are not enough (Fakhry, Peng and Ji, 2016).

This chapter contributes to create new dataset of English text in natural images. This effort is motivated by the lack of the massive amount of annotated training images which should include a complete range of variation and diversity of texts in natural images. Where deep

learning methods require massive completed and varying ranges of training data. Moreover, most of the existing datasets are annotated on the lines or words level, however there is a lack of characters level annotation datasets. The available datasets with character level annotation such as ICDAR03 and IIIT 5K-Word are lack of variation and diversity of various texts in scene images and they are only annotated for text recognition task. The proposed datasets include almost 38,500 samples of English characters and 12,500 words in over 2100 images are annotated by experts. This is a challenging dataset with a good diversity going much beyond previous datasets. Furthermore, the second contribution in this chapter is the proposed augmentation tool which is created to support the proposed dataset due to the missing of augmentation tools for object detection tasks. Where update of the bounding box position is required for object detection augmentation. For this reason, this chapter provide augmentation tool alongside with the proposed dataset to provide bounding boxes augmentation without need to annotate new images, where the position of the bounding boxes and the class can be obtained automatically from the original image. This technique helps to increase the number of samples in the dataset and reduce the time of annotations where no annotation is required.

5.2 Benchmark Datasets

General datasets and relevant evaluation protocols generate robust reference resources for algorithms development and comparison. The datasets and assessment techniques lead to a massive advancement in the fields of scene text detection and recognition (Zhang et al., 2013)(Ye and Doermann, 2015)(Zhu, Yao and Bai, 2016)(Lyu, Liao, et al., 2018).

The existing datasets and their features are summarized in Table 5.1. The following sections will describe in detail some widely used datasets, evaluation protocols and also select some representative image samples from some of the datasets

- **ICDAR 2003 and 2005**: (ICDAR International Conference on Document Analysis and Recognition) They are the first datasets to officially provide a criterion for recognition and detection of scene text.509 completely annotated text images are included in this dataset. 251 of these images are for testing and 258 others are for training. In addition, same set of images was employed in the ICDAR 2005 Text Locating Competition (Lucas et al., 2005).
- **ICDAR 2011 and 2013**: The ICDAR 2011 (Shahab, Shafait and Dengel, 2011) and ICDAR 2013 (Karatzas et al., 2013) Robust Reading Competitions were conducted

with the objective of staying up-to-dated of contemporary progress in recognition and detection of scene text see Figure 5.1. In 2011 and 2013 the datasets were inherited from benchmarks employed in prior ICDAR competitions although with considerable alterations and extensions, as challenges were encountered with the prior datasets (for example, contradictory definitions of 'word' and inaccurate bounding boxes) (Shahab, Shafait and Dengel, 2011).

Twenty-eight video sequences are encompassed in ICDAR '13 datasets and they are arranged to appraise recognition, tracking and text detection of video scenes. Video sequences were acquired with the use of a range of hardware encompassing head-mounted and hand-held cameras, mobile phones from various nations like the United Kingdom, France and Spain. The recognition of text within such video sequences corresponds with particular applications such as seeking a shop in the street or navigating a building (Karatzas et al., 2013)

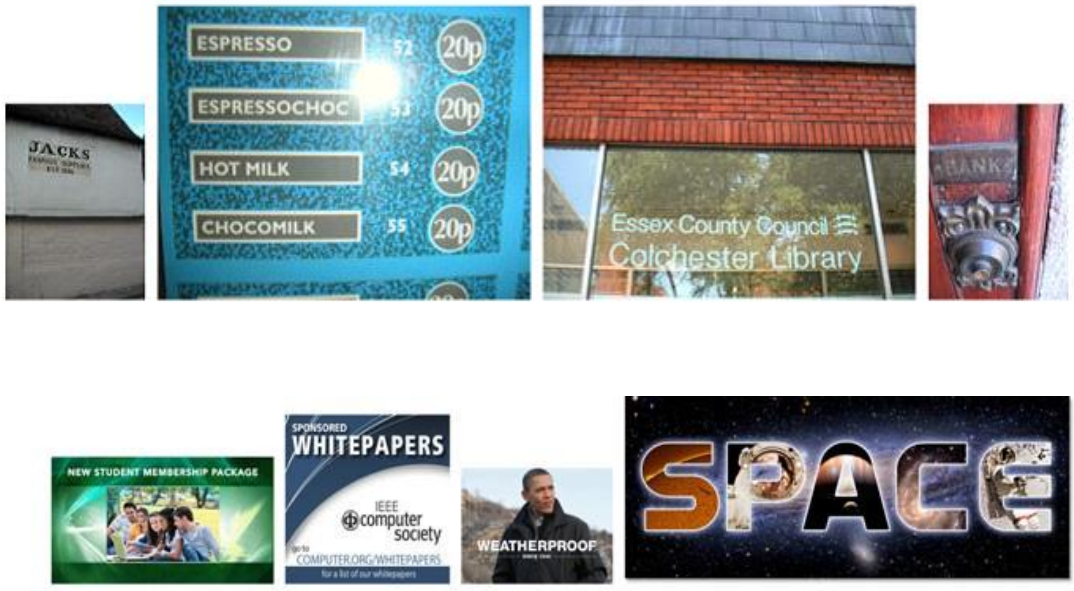


Figure 5.1: : ICDAR 2011 Graphic Text Dataset (Shahab, Shafait and Dengel, 2011)

- **ICDAR 2015:** For this competition which is text reading in the Wild (Zhou et al., 2015) the dataset encompassed natural images acquired from the internet or captured by volunteers amounting to 1000. For testing, 484 images were selected, and for algorithm advancement and validation 500 images were selected. The substance and polygons of all text lines within every image were annotated. The images are real-world natural images with various origins see Figure 5.2 and nearly all the images were taken by novices, the dataset is challenging as well as diverse. The objective of ICDAR 2015

comprises instituting a benchmark to appraise recognition and detection algorithms intended for English and Chinese fonts, and creating a researcher's playground (Zhou et al., 2015).

Figure 5.2 : Images from ICDAR 2015 with Chinese or English scripts(Zhou et al., 2015)

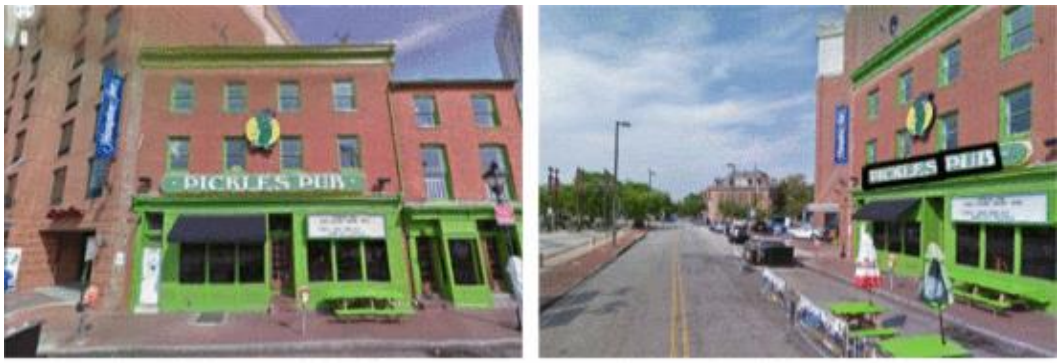
- **Oriented Scene text Database (OSTD):** This dataset was created by (Yi and Tian, 2011) and was employed for the evaluation of text detection algorithms of multi-orientation in natural scenes, It comprises a total of 89 street views, logo and interior scene images.
- **MSRA Text Detection 500 Database (MSRA-TD500):** It comprises a yardstick for the evaluation of detection algorithms for multi-oriented texts within natural settings, which was introduced initially by (Yao et al., 2012). Five hundred images comprising slant and skewed as well as horizontal texts within intricate natural scenes were encompassed within this dataset.
- **The IIIT 5K-word dataset**: All the images were collected from Google by search using Query words such as house numbers, house name plates, billboards, signboard, movie posters. The bounding boxes were used for annotated words in images manually and corresponding ground truth words Figure 5.3. It is consisting of 1120 images and 5000 words. 380 images and 2000 words have been used for training with 740 images and 3000 words for a testing set (Mishra, Alahari and C. Jawahar, 2012). The IIIT5K Word dataset is utilized independently to evaluate character segmentation and/or recognition techniques (Ye and Doermann, 2015) and it provides cropped words (localized text) (Weinman, Learned-miller and Hanson, 2009).

Figure 5.3: Images from IIIT 5K-word Dataset

- **CTW1500 Dataset:** This dataset consists of 1500 images, internet, image library like google Open-Image and phone cameras which have been used to collect images manually with one curve text in each image at least. CTW1500 consist of 10,751 bounding boxes where 3,530 of them are curve bounding boxes. Polygons with 14 vertexes has been used for annotations in CTW1500. The text in CTW1500 dataset basically consists of Chinese and English texts. The dataset include blurred, perspective distortion, indoor, outdoor, born digital images (Yuliang et al., 2017).

- **SynthText Dataset:** synthetic data engine has been used to synthesise natural images with provided texts which is arbitrary in colour, fonts, size, and orientation. The text is provided and aligned to carefully selected image regions in order to have a realism seems. It is consisting of 9 million 32 × 100 images; a 90k word dictionary has been used with equal numbers of word samples. The samples were split in three sets. The testing set consist of 900k of images, in the validation sets , and the remaining for training (Jaderberg et al., 2014).

- **StreetViewText-Perspective**: SVT -Perspective (SVTP) was built from the original SVT dataset. The images in SVTP are selected from side-view images in Google Street View. In contrast with SVT, SVTP images present frontal texts of shop names, street names, etc. Many of SVTP images are taken from non-frontal view angle at the same position on Google Street View where the dataset was built specially for assessing perspective text recognition as shown in Figure 5.4. It is composed of 639 cropped images for testing, each with a 50-word lexicon inherited from the SVT dataset (Phan et al., 2013).



(a) SVT image (b) SVT-Perspective image

Figure 5.4: An image from SVT and the corresponding image from SVT-Perspective. Both images are taken at the same address, and thus have the same lexicon. In (b), the bounding quadrilaterals are shown in black for "PICKLES" and "PUB"

- **SVHN dataset:** The street view house numbers (SVHN) dataset contains various house number regions from Google View images in different countries. It consists of more than 600000 digits where 73257-digit samples are for training, 26032 digits samples for testing, and 531131 others are additional. SVHN dataset is probably used in digit recognition. The images with digit level (number) bounding boxes are shown in Figure 5.5 (Netzer and Wang, 2011).

Figure 5.5: Samples from street view house numbers (SVHN) dataset

- **ICDAR 2017 MLT:** It is consisting of natural scene images with different text styles such as street signs, street advertisement boards, shops names, passing vehicles etc. They are embedded in the images. Different users, different mobile phone cameras have been used to capture dataset images others were selected from Internet. Many of the images include scripts of more than one language in arbitrary orientations hence, it is multi-lingual. It is containing scripts come from 9 languages (Arabic, Bangla, Korean, English, Chinese, German, French, Japanese and Italian) with 7 different scripts.

Furthermore, the dataset contains '"Symbols" script which are the special characters such as + / > :) ' . " - . . It is consisting of 18,000 images 2000 ones per language. The samples were split into three groups 7200 training images, 1800 validation images and 9000 testing images. The annotation is available at word level. Points of four corners have been used to represent bounding boxes, which enable any quadrilateral shape to be represented (Nayef et al., 2018).

- **Total-Text:** (Ch'Ng and Chan, 2018) consists of 1555 images with more than 3 different text orientations: Horizontal, Multi-Oriented, and Curved ones. Text instances in Total-Text are annotated with both quadrilateral boxes and polygon boxes of a variable number of vertexes.

Proper and strong text detection and recognition tasks similar to the other objects detection and recognition models demand massive amount of annotated training images which should be include major variation and diversity of texts in natural images. Furthermore, image aggregation and annotation are big challenge. This leads to lacking appropriate datasets for text detection and recognition using deep learning where the most available datasets such as ICDAR2015, Total-Text, and CTW1500 are lacking the appropriate number of images and the annotation.

Moreover, most of the existing datasets are annotated on the lines or words level and there is a lack of character annotation level datasets. The available datasets with character level such as ICDAR03 and IIIT 5K-Word lack including variation and diversity of various texts in scene images. They are annotated for text recognition task.

Many of the text datasets are built for specific tasks and they are labelled depending on the task that are constructed for them. Annotation methods have been developed from rectangles to flexible quadrangles. ICDAR 2013 and SVT are labelled in horizontal rectangular while MSRA-TD500 labelled with rotated rectangular see Figure 5.6.

Table 5.1 Existing datasets: EN stands for English and CN stands for Chinese, D stands for Detection task and R stands for recognition tasks, Ch stands for characters and W stands for words, BB stands for bounding box annotation

Dataset	Image Num (train/test)	Text Num (train/test)	Orientation	Characteristics	Language	Task	Annotation level
ICDAR03(2003)	258/251	1110/1156	Horizontal	BB	E	D+R	ch+w
SVHN(2010)	73257/26032	73257/26032	Horizontal	House number digits, BB	E	R	digit
SVT(2010)	100/250	257/647	Horizontal	BB	EN	D+R	words
MSRA-TD500 (2012)	300/200	1068/651	M-Oriented	Long text BB	EN,CN	D	lines
IIIT 5K-Word (2012)	2000/3000	2000/3000	Horizontal	BB	EN	R	Ch+w
ICDAR13 Scene Text (2013)	229/233	848/1095	Horizontal	BB	EN	D+R	words
SVTP (2013)	-/639	-/639	M-Oriented	Perspective text, BB	EN	R	word
CUTE (2014)	-/80	-/-	Curved	polygon points of the bounding box	EN	D	line
HUST-TR400 (2014)	400/-	-/-	M-Oriented	Long text BB	EN, CN	D+R	word
ICDAR15 Incidental Text (2015)	1000/500	-/-	M-Oriented	Blur/ Small Defocused	EN	D+R	words lines
ICDARCTW (2017)	8034/4229	-/-	M-Oriented	BB	CN	D+R	Lines
ICDAR17MLT (2017)	9000/9000	-/-	M-Oriented	Script identification	9 language	D	word
Total-Text (2017)	1255/300	-/-	Curved	Irregular Polygon label	EN, CN	D+R	word
CTW (2017)	25K/6K	812K/205K	M-Oriented	Fine-grained annotation	CN	D+R	Ch
CTW1500 (2017)	1000/500	-/-	Curved	Bounding box with *14* vertexes	EN, CN	D	Line level
CASIA-10K (2018)	7K/3K	-/-	M-Oriented	8 coordinates of a quadrilateral	CN	D+R	line-level

Figure 5.6: Example of annotation in MSRA-TD500

ICDAR 2015 and multi-lingual text (MLT) have been labelled by four-point labels. TotalText uses polygon shapes to link ground truth words tightly. Furthermore, it is contained rectangular bounding box Figure 5.7.

Figure 5.7: Example of annotation in Total text dataset

CTW1500 is labelled with horizontal and quadrilateral shape. The annotation methods turned from axis-aligned-rectangle-based methods to rotated-rectangle based, and quadrangle-based methods Figure 5.8.

Figure 5.8: Example of annotation in CTW1500 dataset

5.3 The proposed Dataset

The idea of collecting this dataset is motivated by the lack of datasets that contain arbitrary shapes text (combination of horizontal, multi-oriented, irregular and curved text). The combination of different text orientations (orientation diversity) of text are very common in our real world. Where there is a lake of availability of such data. Furthermore, applying deep learning methods demands huge training data. The massive training datasets are still missing. Moreover, the existing datasets are lacking character level annotation and digit annotation.

Some common datasets such as MSRATD500 (Yao et al., 2012) and the series of ICDAR such as ICDAR 2003, ICDAR 2011, ICDAR 2013 (Karatzas et al., 2013) have great impact on research in the area of text detection and recognition of scene text. It is noted that all the above mentioned ICDARs images consists of horizontal orientation texts (Ye and Doermann, 2015). This led researchers to focus on finding solutions for horizontal scene text detection problem(Yao, Bai, et al., 2013)(Shi, Bai and Belongie, 2017). MSRATD500 new challenge dataset has been provided by Yao et al. (Yao et al., 2012) which consist of texts in multiple orientations. Despite MSRATD500 is known as 'multi-oriented' dataset, the observation emphasizes that most of texts are in straight line mode. Although the curved texts are available around in our daily lives, (home, market, work), researches have not focused on them. CUTE80 (Risnumawan et al., 2014) datasets is the first dataset which existed in 2014 with curved text with only 80 images. The variety of the text is so small. Total-Text emerged in 2017 with a combination of two orientations text instance (Ch'Ng and Chan, 2018), then CTW 1500 come with 3,530 curve bounding box among 10,751 bounding boxes (Yuliang et al., 2017).

The text appear in the scene image in Straight line or curved line where, Straight line: can be stated as a linear function $y = mx + c$, where there is no different angle along the line, In contrast, the curved line points are varying from point to the next point. Horizontal oriented text appears in a sequence of characters which are concatenated by a straight line with bottom alignment; which is same to the multi-oriented text, where character text connected as a straight line with a shift to the horizontal line. The angle of characters in the curved words does not have unified offset (Ch'Ng and Chan, 2018).

5.3.1 Source of Images

Various natural scenes have been used to collect our database such as street signs, shops names, street advertisement boards, the images captured by using cameras of different mobile

phone, websites, and image library such as google open-image library. A large ratio of the images includes text shape in diversity orientation (arbitrary shapes) (horizontal, multi-orientation, curved, irregular). The dataset also includes born, indoor, outdoor images see Figure 5.9.

Figure 5.9: Examples from proposed dataset

5.3.2 Annotation

Two annotation methods have been used to build the ground truth of text in the proposed datasets. It has been decided to annotate character level since there is no available dataset which provides, characters level annotation see Table (5.1). Furthermore, word level annotation is provided in this dataset. Rectangular bounding box and polygon shapes with adaptive number of corner points have been used to annotate text in the proposed datasets see Figure 5.10 and Figure 5.11. the proposed datasets consider English characters and digits where most available datasets aren't considered digits for text detection. Furthermore, word recognition and character recognition annotation are provided by the ground truth. Labelling tool has been built for annotation where the dataset is manually labelled by the researcher. DetEval protocol has been used for evaluation (Wolf and Jolion, 2006).

Figure 5.10*:* Example of annotation of different text orientations (Horizontal, curve, irregular). Word level

Figure 5.11: Example of different text orientations (Horizontal, curve, irregular). Characters level annotation

5.3.3 Data Augmentation

Data augmentation is a method to expand the variety of data in the training sets without the need of gathering more data. The most common methods used for data augmentation are padding, horizontal flipping, cropping, rotation , changing brightness , contrast and adding noise. (Wang, Wang and Lian, 2019)(Mikołajczyk and Grochowski, 2018). Most of the deep learning libraries such as keras (Chollet, 2015), Torch (Collobert, Bengio and Mariéthoz, 2002), and TensorFlow (Abadi et al., 2016) as well as deep neural network such as SSD and YOLO offer augmentation for classification training tasks and it has shown that data augmentations effective in image classification where it is used to reduce overfitting on model. The neural network considers the same image with a slightly different due to applied augmentation to it this help neural network to overcoming the overfitting on the dataset(Wong et al., 2016). However, the support for data augmentation for object detection tasks is still missing. Detection tasks requires to update the bounding box. For this reason, we provide augmentation tool alongside with our dataset to provide bounding boxes augmentation without the need to annotate new images, where the position of the bounding boxes and the class can be obtained automatically from the original image. This technical help to increase the number of images in the dataset and to reduce the time of annotations where no annotation needs.

5.3.4 Image Augmentation for Bounding Boxes

The researcher creates a data augmentation tool which is provided with proposed dataset. Most available augmentation tool for recognition task only does not need to update the position of the bounding boxes after applying transformation. While the augmentation tool for object

detection need to update the position of the objects to get the new positions after applying transformations such as rotation, scaling, shearing. For this reason, we provide augmentation tool alongside with our dataset to provide bounding boxes augmentation without the need to annotate new images. The position of the bounding boxes and the class can be obtained automatically from the original image. The proposed augmentation tool differs from others by applying wide range of transformations which are divided in two groups depending on the object position stability in the image:

- **Invariant Position**

The techniques used for augmentation in this category keep the position of the abject (bounding box) such as adding noise, changing the color palette etc. This form of augmentation changes the pixel values, rather than the pixel positions see Figure 5.12 original images before and after applying lighting, Figure 5.13 adding noise to the original images. Figure 5.14 shows applying contrast to the images. Where the bounding boxes keep their position.

Figure 5.12: Image augmentation by applying lighting

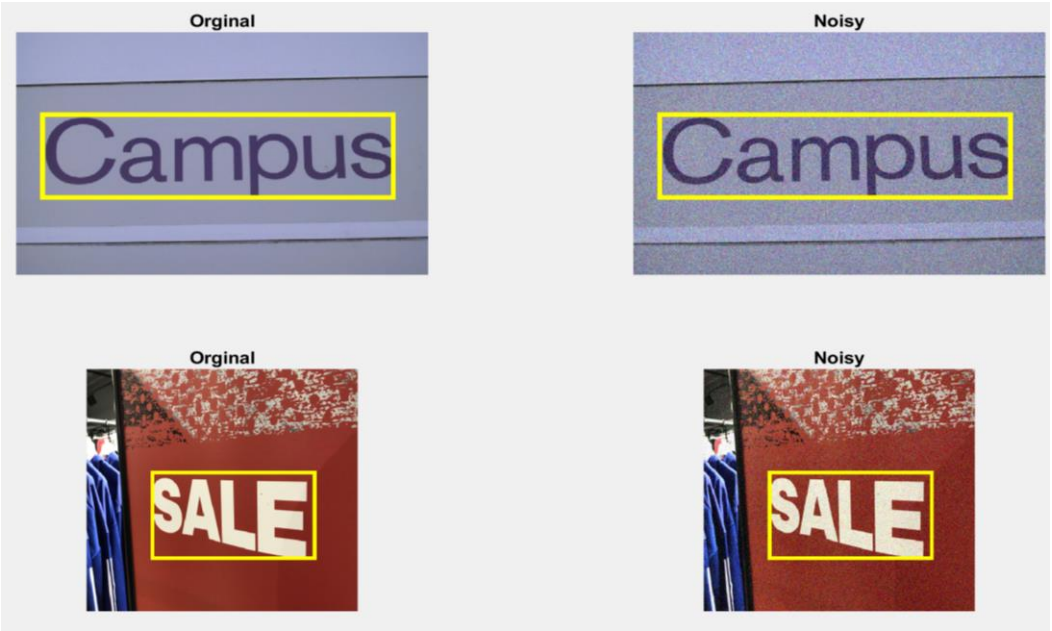


Figure 5.13: Images augmentation by applying noise

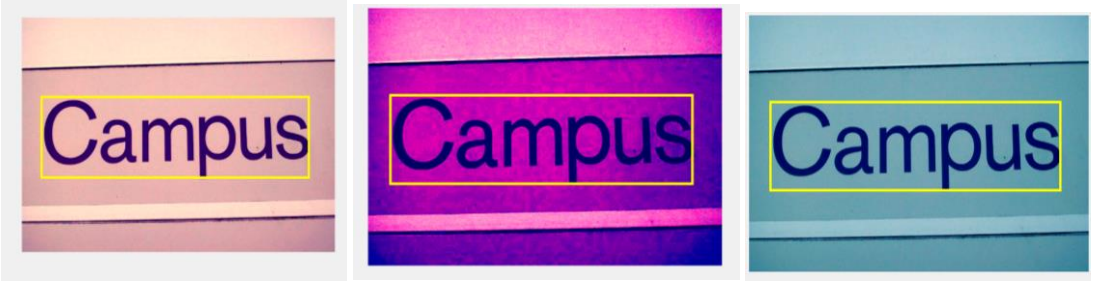


Figure 5.14: Images augmentation by adjust contrast

- **Variant Position**

The techniques used for augmentation in this category modify the position of the object such as rotation, flipping and shear translation. Therefore, the position of the bounding boxes should be modified. To apply this kind of argumentation over bounding boxes, the original position of the boxes and the category saved for next step which is the step of position modification. The coordinates of bounding boxes saved as (x,y) which are the coordinate of the top left, and (x1,y1) which are the coordinate of the right bottom .

Affine Transformation

Transformation such as Scaling, translation, rotation are instants of Affine transformation, it is a linear mapping method that keeps images parallel lines parallel after applying the transformation.

Linear transformation can be represented by matrices. Transformation over images can be represented as a matrix multiplication. If T is a linear transformation mapping Rn to Rm and $\vec{x}$ is a column vector with n entries, then

$$T(\vec{x}) = A\vec{x} \tag{5.4}$$

The usual way to represent a transformation is using a 2 x 3 matrix.

$$A=\begin{bmatrix} a_{00} & a_{01} & b_{00} \\ a_{10} & a_{11} & b_{10} \end{bmatrix} \tag{5.5}$$

Considering that we want to transform a 2D vector

$$X = \begin{bmatrix} x \\ y \end{bmatrix} \tag{5.6}$$

The transformed vector can be obtained by

$$T = A.[x, y, 1]^T \tag{5.7}$$

$$T =\begin{bmatrix} a_{00}x + & a_{01}y + & b_{00} \\ a_{10}x + & a_{11}y + & b_{10} \end{bmatrix} \tag{5.8}$$

Rotation

Rotation transformation is one of the Affine transformations. Rotation transformation of a matrix can be calculated using the rotation angle and the centre coordinates. It can be represented as the equation (5.9):

$$\begin{pmatrix} \alpha & \beta & (1-\alpha).center.x - \beta.center.y \\ -\beta & \alpha & \beta.center.x + (1-\alpha).center.y \end{pmatrix} \tag{5.9}$$

Where

$\alpha = scale * cos$

$\beta = scale * sin$

And θ is the rotation angle

Rotation process changes image size for this reason a modification on the transformation matrix should take into account Figure 5.15 and equations (5.10) (5.11) (5.12) shows the calculation of new dimension.

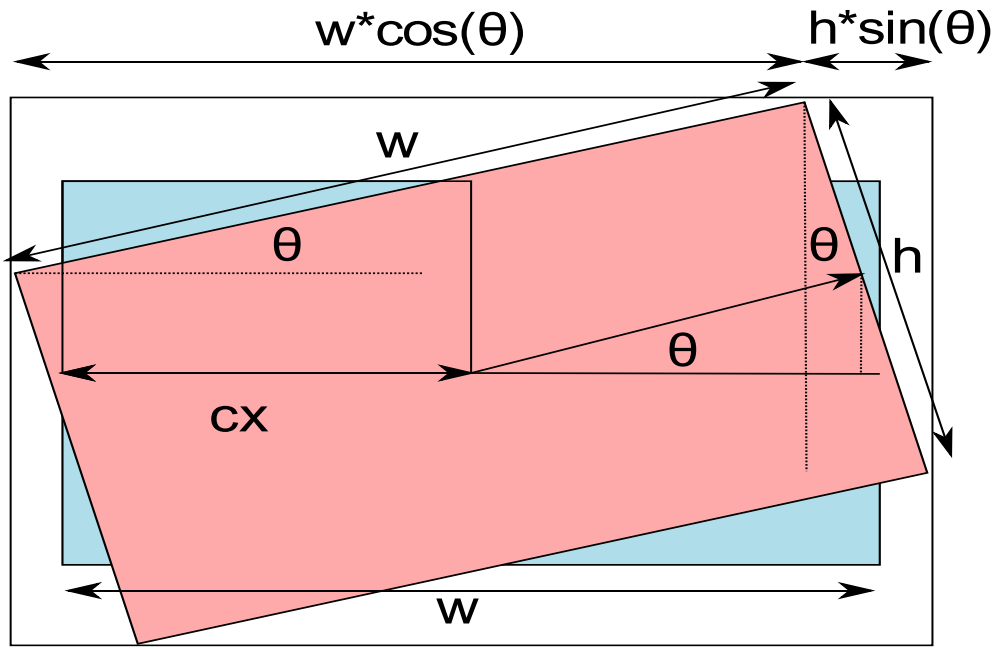


Figure 5.15: Image rotation scheme

The width and height updated by using the following equations

$$new.width = h * \sin(\theta) + w * \cos(\theta) \quad (5.10)$$

$$new.height = h * \cos(\theta) + w * \sin(\theta) \quad (5.11)$$

Rotation transformation affects the image size. This has resulted in modification in the centre of the image and the transformation matrix This is added to the last column of the transformation matrix as follows:(Schneider, Philip K. Eberly, 2003)

$$\begin{bmatrix} \alpha & \beta & (1-\alpha).center.x - \beta.center.y + (new.width/2 - center.x) \\ -\beta & \alpha & \beta.center.x + (1-\alpha).center.y + (new.hight/2 - center.y) \end{bmatrix} \quad (5.12)$$

Rotating the Bounding Box

To rotate the bounding box same. The transformation matrix applied to the 4 corners of the bounding box see Figure 5.16.

Shear

shear transformation or Skewing is a transformation that slants or skew the shape of the object, which could be in two directions X-Shear and Y-Shear. X-Shear transformation keeps the Y coordinate while X coordinates should be relocated, The same transforming is applied for Y-Shear, but the opposite should be applied (Schneider, Philip K. Eberly, 2003) see Figure 5.17, Figure 5.18.

a b

Figure 5.16: Rotating images by a =30 and b= -30

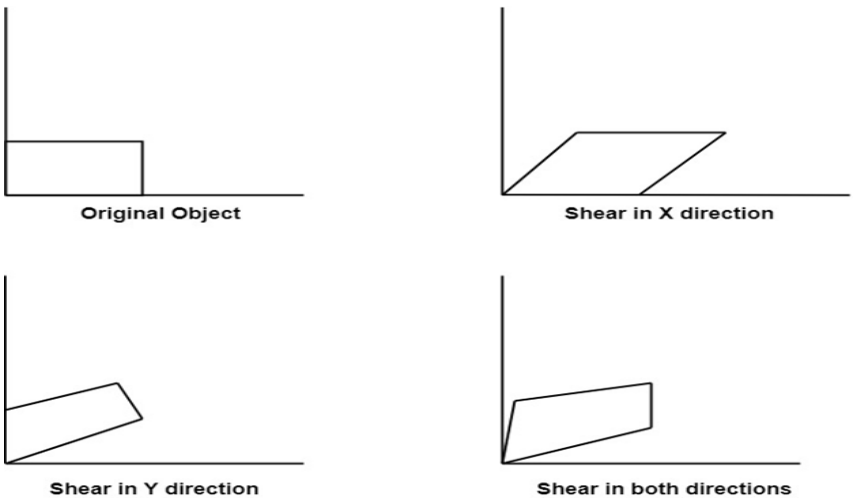


Figure 5.17: Sharing in X,Y, and both direction

The transformation matrix for Shearing in X-Y directions is represented as –

$$X_{sh} = \begin{bmatrix} 1 & Sh_y & 0 \\ sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.13)$$

Where the new coordinates for X-shear can be calculated by :

$$X' = X + Sh_x\, Y \quad (5.14)$$

$$Y' = Y$$

And the new coordinates for X-shear can be calculated by :

$$Y' = Y + Shy.\, X \quad (5.15)$$

$$X' = X$$

Bounding boxes corners should be update by using equation (5.14) for horizontal shear and equation (5.15) for vertical shear.

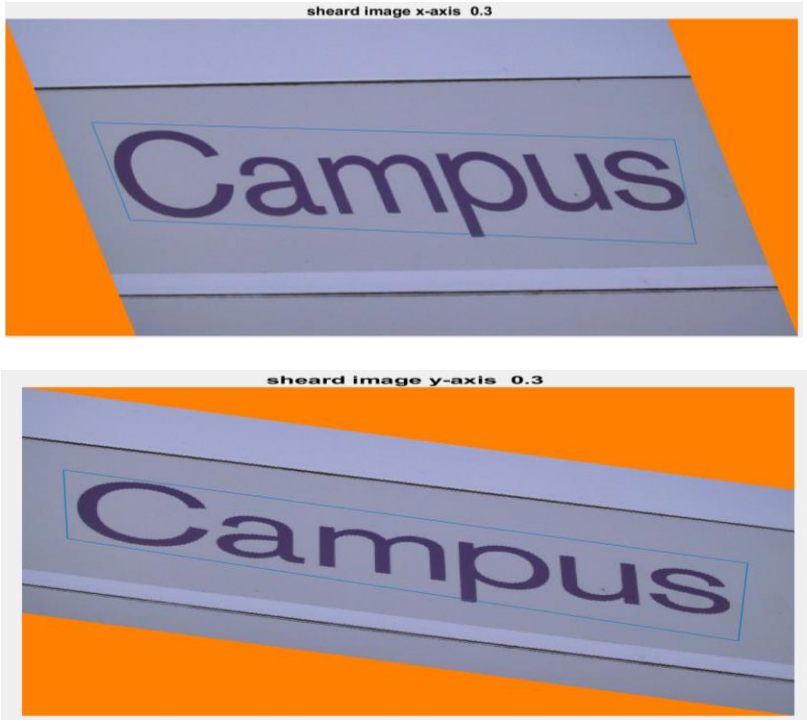


Figure 5.18: Apply horizontal and vertical shearing

Scale

Scaling is the method of modifying an image to larger (scale > 1.0) or smaller (scale < 1.0). To scale image to 25% of it is existing size a factor of a decimal percent equal to 0.25 should be uses. The same scale should be applied to the bounding boxes coordinates to modify their positions Figure 5.19.

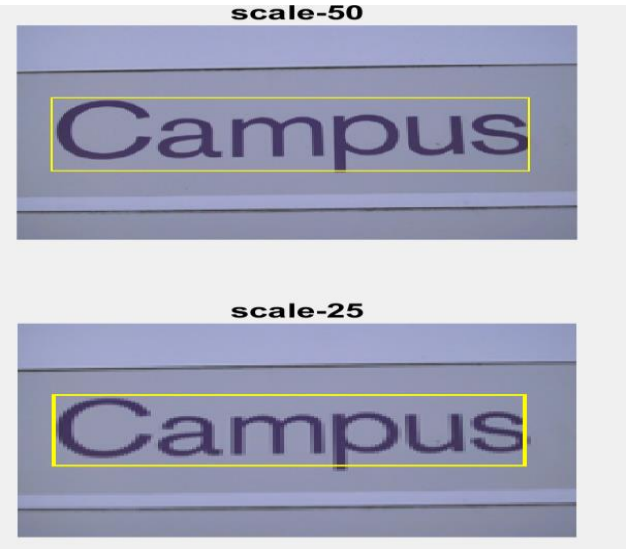


Figure 5.19: scale image by 0.5 and 0.25

The Data Augmentation Tool

The data augmentation tool has been built using Matlab R2019a. This tool helps users to increase the samples in the dataset automatically by using commands available in the tool interface. Figure 5.20 shows that the augmentation type can be selected from two types of available augmentation which are invariant position (lighting, noising, and contrast) and variant position (rotation, shearing, and scaling). By selecting the type of augmentation, it will appear a new interface screen Figure 5.21. The tool provides users with facilities to choose the range of parameters and images to apply such as selecting rotation. Figure 5.21 demonstrates that the user can also select the number of angles and images to apply the chosen augmentation then save the results to the dataset.

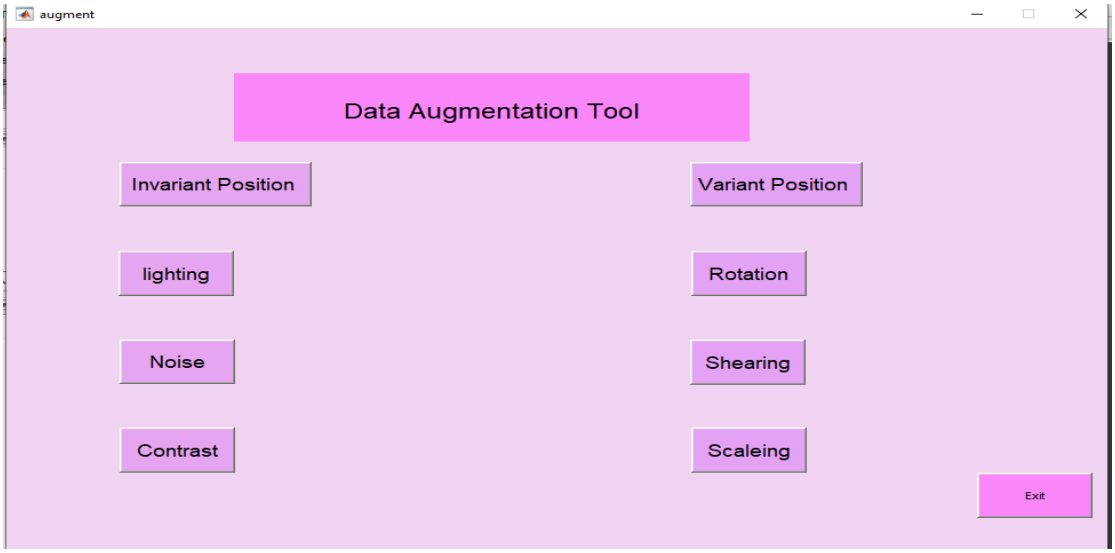


Figure 5.20: data augmentation tool Interface

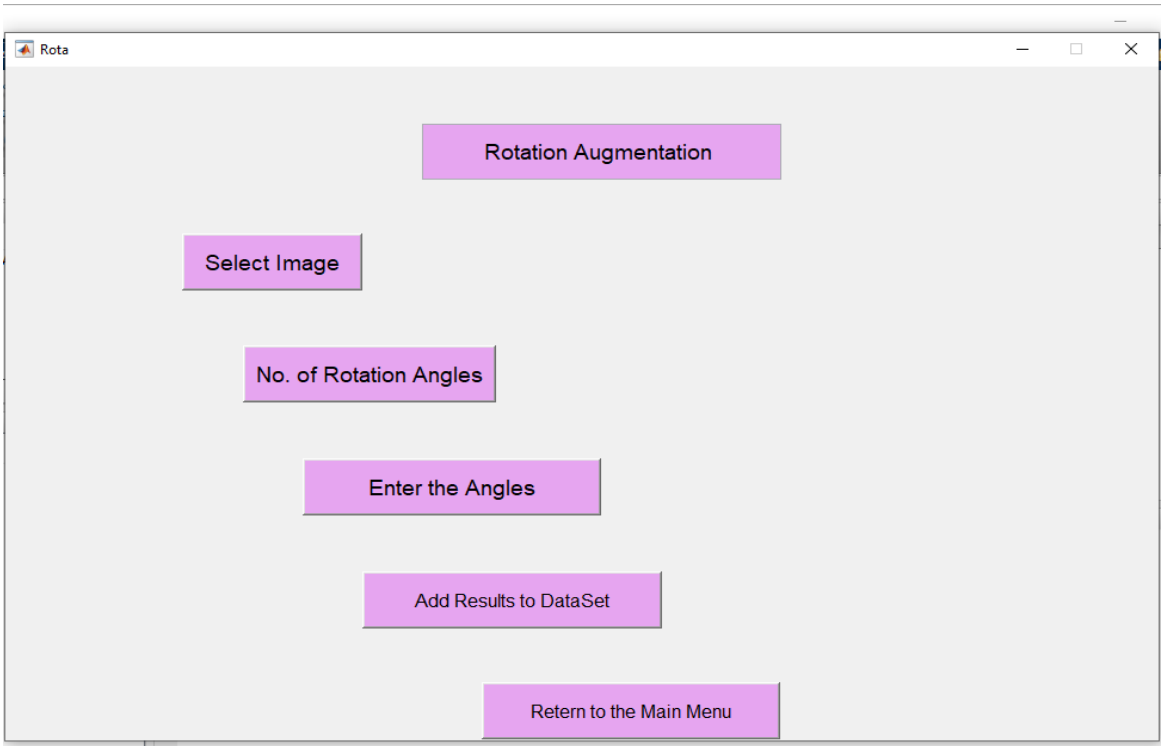


Figure 5.21: Interface of rotation augmentation

5.4 Conclusions

This chapter proposed two new contributions on the field of text detection and recognition. Arbitrary text is available in our real life but currently few datasets and techniques target at arbitrary text detection and recognition. Furthermore, most of the available datasets suffer from the lack of samples for training, especially those used for deep learning approaches. Moreover, the existed dataset lack diversity and variation texts, where most of datasets are specialized in one orientation of texts such as horizontal, multi orientation or curve. They are annotated in the line or word level where character level annotation is not available especially for detection task.

The above limitation encourages the researcher to propose a novel dataset which consist of 2100 images, datasets annotated in the word level and character level with 38500 characters and 12,500 words. Furthermore, we use bounding boxes and polygons for annotation. The proposed dataset is various. It consists of arbitrary shapes texts (horizontal, multi orientation and curve texts). To fill the gap in the limitation amount of the training samples. Data augmentation tool has been proposed alongside with the proposed dataset. This tool works on the level of bounding boxes which is not available, and it is crucial for texts detection. The proposed tool produces two types of augmentation which are invariant position where the

position of the bounding boxes keeps their position such as adding noise, lightning and contrast and variant position. The position of bounding boxes should be located in new position depending on the augmentation transform that applied to the original image. This tool helps to increase the number of training samples without the need to annotate new images which are created by using the proposed augmentation tool.

Chapter 6

End-to-End Text Detection and Recognitions Model

6.1 Introduction

The traditional methods based on machine learning, which extract hand-crafted features, consist of multiple steps. These kinds of methods result in complicated and inefficient detection and recognition models, which lead to accumulations of errors. In contrast with hand-crafted feature methods, deep learning methods have the ability to learn high-level features robustly and can outperform traditional methods in accuracy and efficiency (Long et al., 2018) .

Recently, deep neural networks have been used widely for text detection and recognition. These methods have dominated the area of detection and recognition by producing good results. Most of the earlier methods are strict on detecting word-level or line-level bounding boxes. Those methods have the limitation of detecting text in an arbitrary shape, as well as curved and blurred texts (Long et al., 2018).

Recently, most proposed methods have shown high performance of training to detect texts of word level (Lyu et al., 2018)(Liu et al., 2018)(Deng et al., 2018)(He et al., 2018)(Long et al., 2018)(He et al., 2017)(Hu et al., 2017)(Shi, Bai & Belongie, 2017).There are many limitations when it comes to these methods, such as long, curved, and deformed texts which cannot be detected.

The alternative method, which is character level, has many advantages, e.g. it can detect characters of arbitrary shape, multi-orientation, and those which are curved. It does not need a dictionary to recognise them. Furthermore, most previous methods deal with text detection and recognition separately, where the result of detection model fed to the recognition model, both models are extremely integrally and correlated. Dealing with them separately might result in unpromising performances for both models (Hu et al., 2017).

Recently, text spotting models, which are end-to-end methods, have been proposed for localising and recognising text in a unified network. Those methods need relatively complex training procedures. Methods such as those from Bušta, Neumann and Matas (2017) and Li, Wang and Shen (2017) have some limitations when it comes to dealing with text spotting, such

as the inaccurate location of the detection in the early iterations, which affects the recognition phase. Moreover, those methods are not capable of spotting curved text.

This chapter proposes a new method for text detection and recognition, which attempts to overcome the previously-mentioned limitations by detecting and recognising each character individually. The proposed framework is an end-to-end character detection and recognition system designed with an improved SSD convolutional neural network, where inceptions layers are added to the SSD networks and the aspect ratio of the characters is considered different from other objects. Compared with other methods, the proposed method is capable of detecting and recognising characters using the end-to-end model completely. The proposed network has the ability to spot digits alongside characters. To the best of my knowledge, this is the first method to integrate text and digits detection and recognition; all of the existing methods focused on either texts or digits, although both are available in scene images together. The proposed method has the ability to spot scene text of an arbitrary shape (horizontal, oriented, and curved).

6.2 Overview of Proposed Method:

6.2.1 SSD: Single Shot MultiBox Detector

Researchers from Google published a SSD deep learning architecture in 2016. They combined the regional proposals network and feature extraction to introduce a single deep neural network for object detection. Various scales and aspect ratios have been used over a group of default boxes and applied to the feature maps. An image classification network has been used to calculate feature maps in a single step over bounding boxes (Liu et al., 2016).

- **SSD Model**

The SSD architecture is built up of three main components, namely the base network, auxiliary feature layers (multi-scale feature maps structure), and the prediction layers. The feed-forward convolutional network has been used as a foundation of the SSD. A standard architecture for high-quality image classification has been employed to build the first network layers Figure 6.1. These are called base network (VGG-16 has been used here as the base network, while other networks can also be used) of the SSD before any other classification layers(Liu et al., 2016).

The VGG-16 architecture was selected to be employed as the base network due to its robust performance in the tasks of image classification, and SSD modifying the structure of VGG-16 by converting the original fully connected layers via adding a collection of

auxiliary convolutional layers (from conv6 onwards) Figure 6.2. Those added layers make it possible to extract features on various scales and gradually decrease the size of the input in each subsequent layer. The base network output is a feature map with a size of 19 * 19 * 1024.

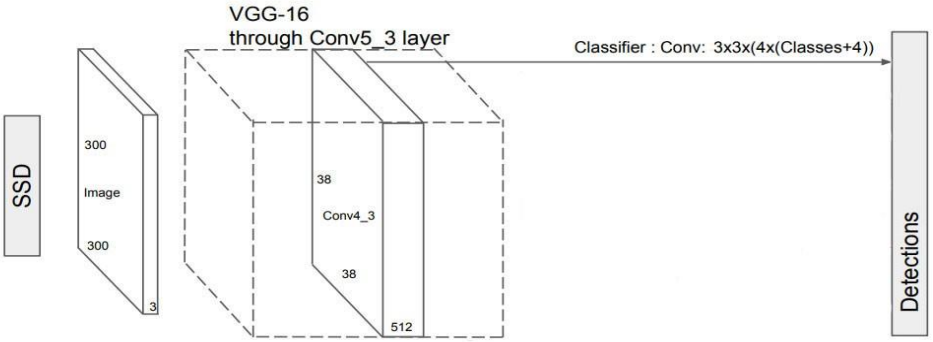


Figure 6.1: First part of SSD which is adopted from VGG-16.

Following this, a multi-scale feature maps structure was added to the network for the purpose of detection. Four additional convolutional feature layers were added to the end of the base network. The size of the feature map layers was decreased gradually to produce detection forecasting on multiple scales until it reached a size of 1 * 1 * 256 as a final feature map (Ning et al., 2017).

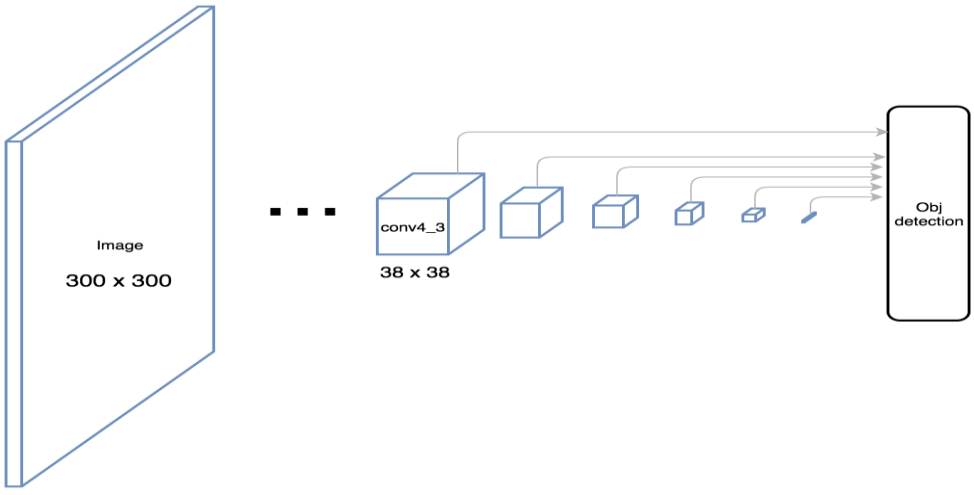


Figure 6.2: Second part of SSD.

Each convolutional feature layer which is present in the base network or added feature layers has the capability to generate a fixed collection of detection forecasts by utilising a group of convolutional filters Figure 6.3.

One of the important parts of the SSD is the prediction layers. The SSD approach generates a set of fixed-size bounding boxes and scores, those boxes containing object class instances, then non-maximum suppression technique is used to provide the final detections. Many feature maps representing multiple scales are utilised rather than using one map for predicting. The layers conv4_3, conv7, conv8_2, conv10_2 and conv 10_2 were applied for the bounding box predictions(Liu et al., 2016).

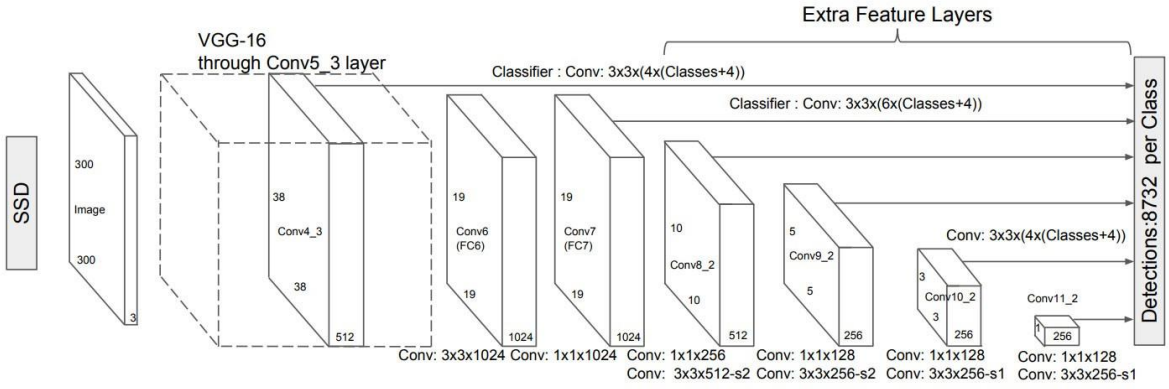
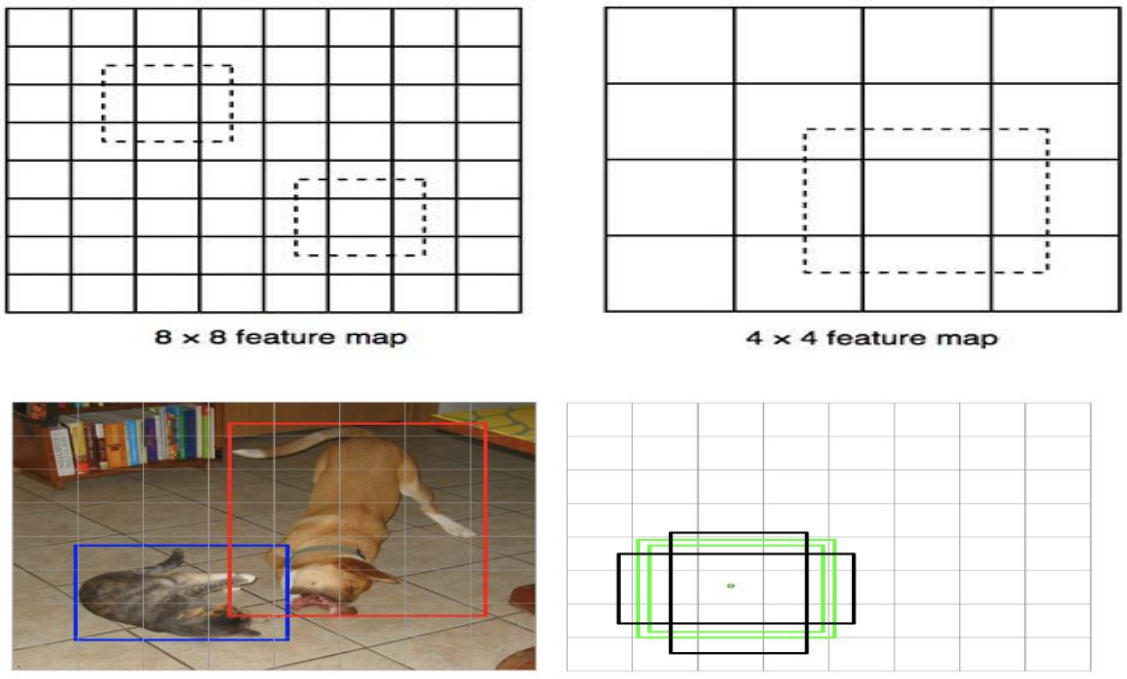


Figure 6.3: The SSD network architecture (Liu et al., 2016).

The default boxes were used to accurately clarify the chosen bounding boxes depending on their sizes, aspect ratios and positions through the image. The SSD employs 8,732 default boxes with the goal of determining the exact default boxes to be used for a given image and predicting offsets from the selected default boxes to gain the final prediction.



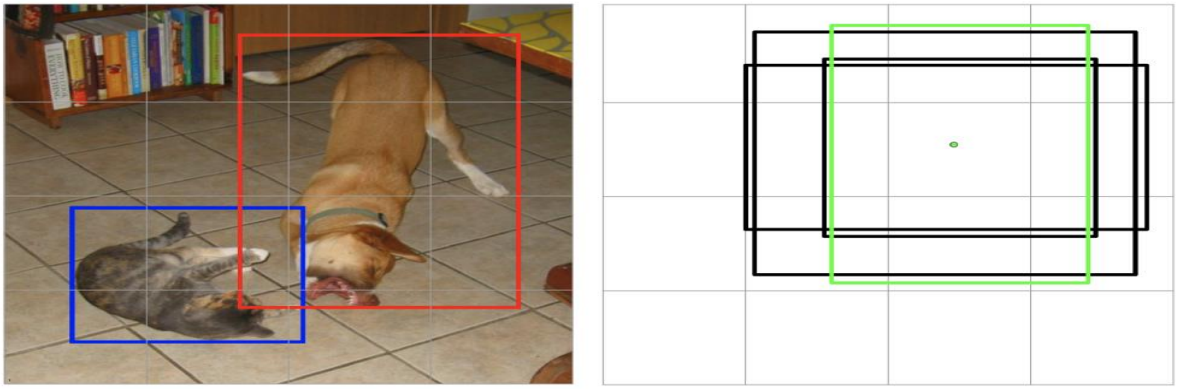


Figure 6.4: Bounding boxes for different scales and aspect ratios (Szegedy et al., 2015).

In Figure 6.4, there are two feature maps with resolutions of 8 * 8 and 4 * 4, where the first includes the default bonding box which is fit to the bonding box corresponding to the cat, while the second one matches the dog. Figure 6.4 shows that the green boxes are the positive default boxes that match one of the ground truth boxes (cat, dog), while the negative default boxes (black) do not match any of the ground truth boxes.

The main task of the trained SSD method is to classify defaults boxes to find which are the positive boxes from the 8,732 SSD boxes, and find the offsets of those default positive boxes' coordinates, to gain the final bounding boxes (Szegedy et al., 2015).

- **Select default boxes**

The SSD depends on default boxes, where selecting scales and the aspect ratios is crucial. The default boxes are designed for each feature map corresponding to a specific scale of the default boxes along with a list of aspect ratios for each scale.

Assume that the m feature maps are needed for prediction, then the scale of the default boxes for each feature map is computed as:

$$s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{m - 1}(k - 1), \quad k \in [1, m] \tag{6.1}$$

where minimum scale *s_min*, means that the lowest scale is set to 0.2 and maximum scale *s_max* the highest scale set to 0.9 all other layers are spaced regularly in between those two parameters.

Default boundary boxes are selected manually. SSD determines a scale value for every feature map layer. Starting from the left, Conv4_3 detects objects at the smallest scale 0.2 and then raises linearly to the rightmost layer at a scale of 0.9. Joining the scale value with the target aspect ratios.

The default boxes' heights and widths for feature maps corresponding to a specific scale are obtained by using equations (6.2) and (6.3):

$$w_k^a = S_k\sqrt{a_r} \quad (6.2)$$

$$h_k^a = \frac{S_k}{\sqrt{a_r}} \quad (6.3)$$

where a_r constitutes the aspect ratios for the default boxes, $a_r \in \{1,2,3, 1/2, 1/3\}$; furthermore, an additional default box has been added with aspect ratio =1, and it is a scale calculated according to equation (6.4). This means that there are six default boxes per feature map location (Szegedy et al., 2015).

$$S'_k = \sqrt{S_k S_{k+1}} \quad (6.4)$$

6.2.2 Inception

Deep neural networks are different from traditional methods which have multi-part and complex structures where the performance of those networks can be enhanced by increasing their depth and width. To make the networks stronger and increase the ability of extracting features, the networks should go deeper. Furthermore, the objects which appear in the image vary considerably in size; some of them are very small, while others are huge. In this case, to detect all objects, selecting the right kernel size is extremely crucial. To detect large objects, a larger kernel can be used, while a small kernel is effective for small objects. This leads to increased network size, thus raising the number of parameters and competition time, which results in overfitting (Ning et al., 2017).

Inception has been introduced to solve the above problems and to strike a compromise between performance and speed. According to Szegedy et al. (2015), modifying all or half of the convolution layers to sparse links can help in solving the problem of parameter rising. Thus, as mentioned, inception has been introduced to solve the above problems and to strike a compromise between performance and speed; this is possible due to the ability of inception to block and to catch more information without increasing networks' complexity (Ning et al., 2017). The structure of the SSD consists of the VGG network and extra layers added after VGG layers (see Figure 6.3). The week point in those added layers is that they have one type of convolutional kernel ((3 × 3) kernel); this led to the extraction of only specific sizes of objects. According to Sermanet et al. (2013) and Girshick (2015), the different object scale problems can best be solved by the processing of images of multiple sizes and then merging the results. Furthermore, Luo et al. (2017) clarified that the resolution of the feature maps reduced as much

as adding new convolutional layers; for each layer the receptive field of the feature maps should continue increasing. For this reason, the smaller objects detected at the earlier layers have a smaller receptive field, while larger objects can be detected by the layer which has a larger receptive field. This encourages us to adopt the inception structure, which modifies the last layers of the SSD by replacing them with convolutional layers with different kernel sizes.

We introduced the inception structure in extra layers after VGG16 by increasing the type of convolution kernels. Therefore, the receptive field range is expanded. This led to increases in the model's sensitivity to small objects without losing large objects.

Instead of the last layers of the SSD, the inception block was used, where different kernel sizes of convolutional layers are stacked, and those layers come with different receptive fields. Where 1×1 convolutional layers, 3×3 convolutional layers and 5×5 convolutional layers to replace the extra original 3×3 convolutional layers Figure 6.5. This helps to keep more object details (Szegedy et al., 2015).

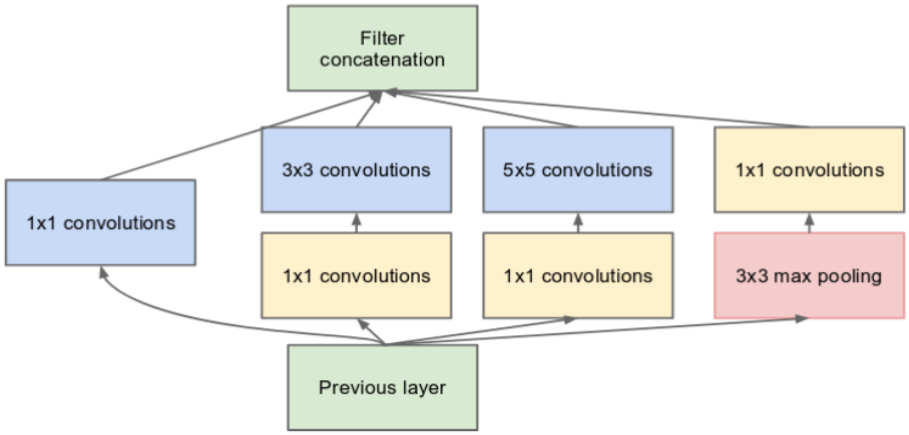


Figure 6.5: Original inception module (Szegedy et al., 2015).

Replacing 5×5 convolution with two 3×3 convolution layers helps to reduce the cost of the calculation (Figure 6.6). The cost of a 5×5 convolution is 2.78 times more than a 3×3 convolution, hence this replacing helps to improve the performance (Szegedy et al., 2015).

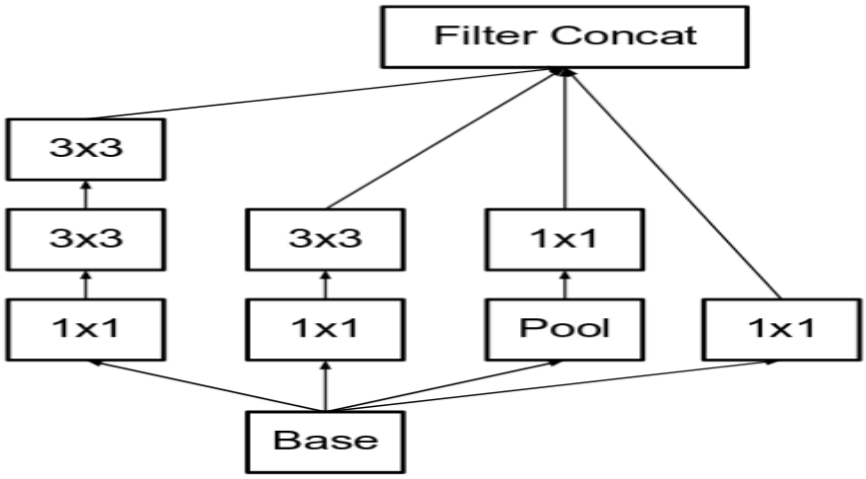


Figure 6.6:5 × 5 convolution is replaced by two 3 × 3 convolutions (Model A) (Szegedy, Vanhoucke, Ioffe & Shlens, 2016).

To reduce the cost and make it cheaper by 33%, the replacement of convolutions of filter size n×n to a group of 1×n and n×1 convolutions was used Figure 6.7, where a 3×3 convolution is equal to the performance of a 1×3 convolution, and then performs a 3×1 convolution as its output.

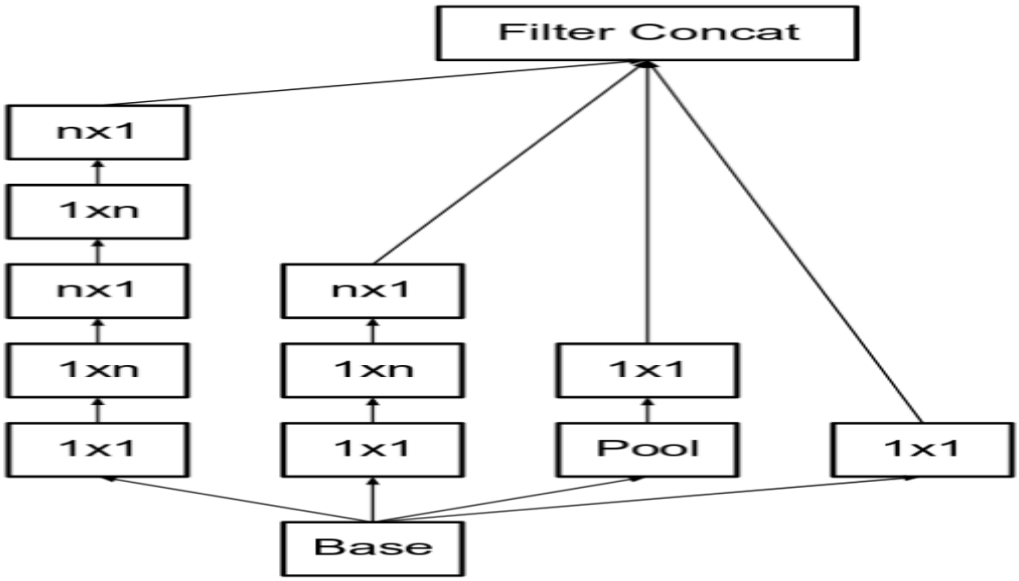


Figure 6.7: Model (B) replacing the n × n convolutions. n = 7 (Szegedy, Vanhoucke, Ioffe & Shlens, 2016).

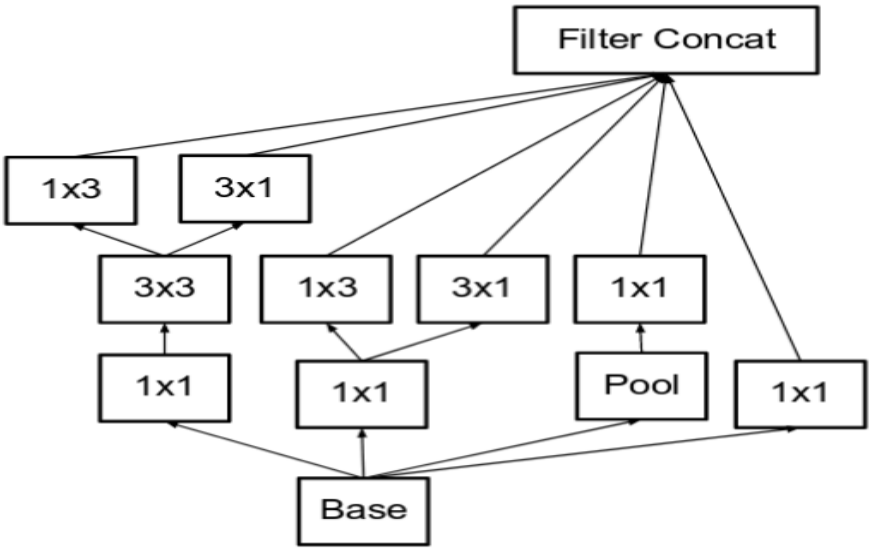


Figure 6.8: Module C wider inception (Szegedy, Vanhoucke, Ioffe & Shlens, 2016).

The expansion of filter bank has been proposed in inception modules. This structure is usually used on the 8 × 8 grids to boost high dimensional representations. This architecture has been proposed for use only on the coarsest grid, which producing high dimensional sparse representation. This idea has made the structure wider instead of deeper. This would be excessive minimisation of dimensions, and hence lead to a loss of information (see Figure 6.8) (Szegedy, Vanhoucke, Ioffe & Shlens, 2016).

6.2.3 The Proposed structure

The three models (A, B, C) are applied to construct new layers, which replace the extra layers after VGG16 in the SSD network. Three layers of model C have been added, followed by three inceptions of model B, and then four inception layers of model A. Table 6.1 illustrates the architecture of the proposed network. The structure of the network is the following:

First and Second Layers:

The input for the proposed model is a 300×300×3 RGB image which passes through the first and second convolutional layers with 64 feature maps with filters measuring 3×3 and pooling with a stride of 1. This results in the image dimensions changing to 300x300x64. Following this, a maximum pooling layer or sub-sampling layer is applied with a filter measuring 3×3 and a stride of 2. This will reduce the image dimensions to 150x150x64.

Third and Fourth Layers:

Two convolutional layers are applied with 128 feature maps measuring 3×3 and a stride of 1. Then there is a maximum pooling layer with filter size 3×3 and a stride of 2. This layer is the same as the previous pooling layer, except that it has 128 feature maps. So, the output will be reduced to 75×75×128.

Fifth and Sixth Layers:

The fifth and sixth layers are convolutional layers with filter size of 3×3 and a stride of one. Both use 256 feature maps. The two convolutional layers are followed by a maximum pooling layer with filter size 3×3, a stride of 2, and 256 feature maps. This reduces the image dimensions to 38×38×256.

Seventh to Ninth Layers:

Next there are three inception layers of model C, followed by a maximum pooling layer. All convolutional layers have 512 filters of size 8×8 and a stride of 1. The final size will be 38×38×512.

At this stage, 5776 of the default boxes will be provided here. Furthermore, confidence and localisation losses are also provided here.

Tenth to Twelfth Layers:

The next three layers are model B inception layers with filters of size 7×7 and 512 feature maps. The final size will be 19×19×512.

Thirteenth to Fourteenth Layers:

Two of the fully connected layers are used to generate 2,166 default boxes with localisation, confidence losses, and position.

Fifteenth to Sixteenth Layers:

Two of model A's inception layers where the filters are of size 5×5 with 512 feature maps. The final size will be 10×10×512. 600 boxes are provided here.

Sixteenth to Seventeenth Layers:

Two of model A's inception layers where the filters are of size 5×5, with 256 feature maps. 150 boxes will be generated here.

Eighteenth to Nineteenth Layers:

Two convolutional layers with filter size 3×3 and strides of 1 and 2, in addition to 128 and 265 feature maps respectively are used to generate four bounding boxes.

Table 6.1: Summary of proposed Architecture.

Layers type	Feature Maps (filters)	Size	Kernel Size	Stride	Activation
Image	1	300×300×3	-	-	-
Convolution 2D	64	300×300×64	3×3	1	Relu
Convolution 2D	64	300×300×64	3×3	1	Relu
Max pooling	-	150×150×64	2×2	2	Relu
Convolution 2D	128	150×150×128	3×3	1	Relu
Convolution 2D	128	150×150×128	3×3	1	Relu
Max pooling	-	75×75×128	2×2	2	Relu
Convolution 2D	256	75×75×256	3×3	1	Relu
Convolution 2D	256	75×75×256	3×3	1	Relu
Convolution 2D	256	75×75×256	3×3	1	Relu
Max pooling	-	38×38×256	2×2	2	Relu
Inception model C	512	38×38×512	8×8	1	Relu
Inception model C	512	38×38×512	8×8	1	Relu
Inception model C	512	38×38×512	8×8	1	Relu
Box generator stage 1 5,776 boxes will be generated with localisation, confidence, and positions					
Max pooling	-	19×19×512	2×2	2	Relu
Inception model B	512	19×19×512	7×7	1	Relu
Inception model B	512	19×19×512	7×7	1	Relu
Inception model B	512	19×19×512	7×7	1	Relu
Max pooling	-	19×19×512	2×2	2	Relu
FC	1024				
FC	1024				
Box generator stage 2 2,166 boxes will be generated with localisation, confidence, classes, and positions					
Inception model A	256	19×19×256	5×5	1	Relu
Inception model A	512	10×10×512	5×5	2	Relu

Box generator stage 3 600 boxes will be generated here					
Inception model A	128	10×10×128	5×5	1	Relu
Inception model A	256	5×5×256	5×5	2	Relu
Box generator stage 4 150 boxes will be generated here					
Convolution 2D	128	5×5×128	1×1	1	Relu
Convolution 2D	256	3×3×256	3×3	2	Relu
Box generator stage 5 36 boxes will be generated here					
Convolution 2D	128	1×1×128	1×1	1	
Convolution 2D	256	3×3×256	3×3	2	
Box generator stage 6 4 boxes will be generated here					

The total number of boxes can be calculated by adding the boxes provided at each stage= 5776+2166+600+150+36+4=8732.

One of the most important parts of the structure is the default boxes generation layer. The size of the feature map of the previous convolutional layer is the input for this layer. The boxes generated depend on the scales and ratios. The coordinates and sizes of all default boxes are generated by the box's generator layer for the specific feature map.

The prediction consists of the following:

If the N_{class}= 62, the final prediction should have the size of 62+1+4+8.

62: N_{class} confidence scores.

1: Background class confidence score in case of a negative box.

4: Predicted position offsets for the box (X_{left} , X_{right}, Y_{top} ,Y_{bottom}).

8: 4 positions plus 4 variances. Those are the positions generated by the generator layer.

6.2.4 Improve the Aspect Ratio

In the SSD at each feature map position default boxers have different aspect ratios (ar) where the aspect = {1,2,3,1/2,1/3}. The characters ratio is different from that of general objects.

Most letters of the English language have an aspect ratio of close to 1. For this reason, we defined six aspect ratios for default boxes that include 1/2, 1/3, 1/5, 1/4, 1, 1.5. An example is shown in Figure 6.9.

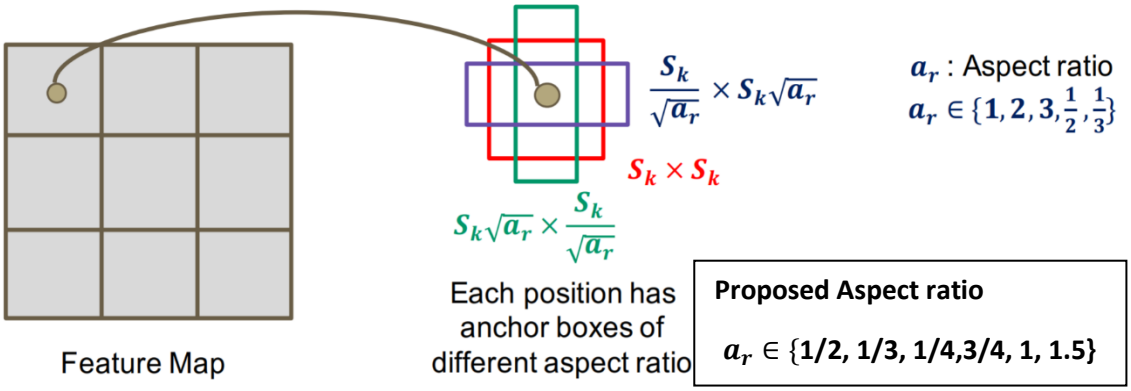


Figure 6.9: The proposed aspect ratio.

6.2.5 Training

To train the SSD, the ground truth of the dataset should be available and allocated to the specific outputs in the group of detector outputs. Furthermore, to start training the detection scales, the collection of default boxes and the data augmentation technique should be selected. Once those parameters have been specified, the back propagation and loss function are utilised end-to-end. MultiBox loss (Erhan et al., 2014)(Szegedy et al., 2014) has been used as a loss function which is composed of two terms: the localisation loss and the confidence loss. After training, the SSD produces a prediction (C+4) for each of the 8,732 boxes, with C class scores and four localisation offsets.

The task of the training phase is matching all the default boxes to the ground truth boxes by using Jaccard overlap (IoU) (Erhan et al., 2014)(Szegedy et al., 2014). The matching technique in the SSD is slightly different from MultiBox (Erhan et al., 2014), where MultiBox matches one default box with maximum overlap with ground truth, but the SSD uses a threshold of 0.5 to match multi-default boxes with ground truth. Thus, the SSD predicts multiple overlapping with high scores instead of selecting just one default box with maximum overlap.

For this reason, SSD training is modified to suit the multi-object classes to explain this modification. The parameter $x_{ij}^{p} = \{1,0\}$ is used to refer to the matching between the i_th default box and the j_th ground truth box of class p.

Loss function

The **objective loss** is the summation of the localisation loss (loc) and the confidence loss (conf), as shown in equation (6.5):

While **localisation loss** consists of the negative boxes which are the default boxes that mismatch to any ground truth boxes, the matched boxes, which are considered as positive, add to both confidence loss and localisation loss:

$$L(x, c, l, g) = \frac{1}{N\left(L_{conf}(x,c) + \propto L_{conf}(x,c,g)\right)} \qquad (6.5)$$

N represents the number of matched default boxes; when there is no matched N = 0, then loss = 0. The relation between the predicted box (L) and the ground truth box (g) represents localisation loss, which is a smooth L1 loss between the real offsets of the default boxes to the ground truth boxes and the predicted offsets.

$$L_{loc}(x, l, g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx,cy,w,h\}} x_{ij}^{k} \text{smooth}_{\text{L1}}(l_i^m - \hat{g}_j^m)$$

$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^w \qquad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h$$

$$\hat{g}_j^w = \log\left(\frac{g_j^w}{d_i^w}\right) \qquad \hat{g}_j^h = \log\left(\frac{g_j^h}{d_i^h}\right)$$

……(6.6)

where $\widehat{g_i^m}$ is the target offsets

l_i^m the predicted offsets

g_j^m the ground truth box coordinates

d_i^m default box coordinates

$x_{ij}^m = \{0,1\}$ Indicator for default box i match to ground truth box j of class p. It is =1 when IoU> 0.5 between default box i match to ground truth box j.

Then the **confidence loss** calculated as it is the softmax loss over multiple classes

$$L_{conf}(x, c) = -\sum_{i \in Pos}^{N} x_{ij}^p log(\hat{c}_i^p) - \sum_{i \in Neg} log(\hat{c}_i^0) \quad \textbf{where} \quad \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)}$$

…(6.7)

To make the training phase quicker and more stable, the SSD uses the ratio 3:1 to balance between positive and negative boxes, and this is due to the large number of negative boxes. As

soon as the matching is finished, the negative boxes are sorted by using their confidence loss to pick those with the highest loss, which are kept.

6.2.6 Non-Maximum Suppression (NMS)

NMS was used to eliminate the duplicate predictions referring to the same object. Predictions were sorted according to the confidence scores. NMS with IoU threshold of 0.45 was conducted to produce the final characters. SSD starts from the top confidence prediction to check the previously-predicted boundary boxes which have an IoU higher than 0.45 with the current prediction for the same class. When it exists, then current prediction will be ignored (Szegedy et al., 2014).

6.2.7 Characters Grouping

To group individual characters into words, the method of Baek (2019) was adopted. This technique deals with curved texts effectively. The character regions are scanned in a specific direction to locate the local maxima line of the specific region. Figure 6.10 shows the maxima line in blue arrows. The length of those lines should be considered to prevent obtaining an uneven polygon; the maximum length of all lines should be equal to the maximum length among them. Furthermore, the centre line is the line connecting all the centre points of the local maxima lines – those lines shown in Figure 6.10 in yellow. The slope angle of the characters is considered by rotating the local maxima lines to be perpendicular to the centre line shown in red arrows in Figure 6.10. Following this, for each local maxima line we calculate the endpoints, which are considered as candidate control points for the polygon. Then the most outer local maxima lines shifted to out along the local maxima centre line, this process will update the final control points (green dots). This process helps to fully cover the region of text (Baek et al., 2019).

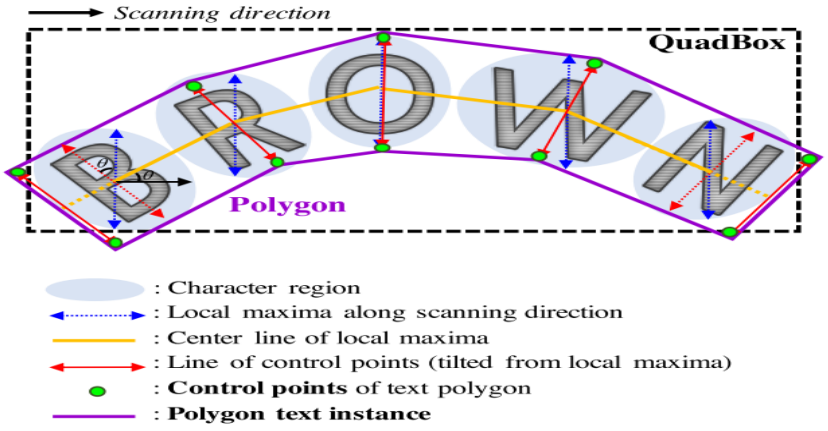


Figure 6.10: Polygon generation for arbitrarily shaped texts (Baek et al., 2019).

6.2.8 Implementation Details

The proposed method has been trained with 300 × 300 images using stochastic gradient descent (SGD). Momentum was set to 0.9 and weight to 0.00004. The initial learning rate was set to 0.004. The proposed approach is trained on the proposed datasets' character annotation level for 50,000 iterations, following the scheme in Liu et al. (2016). All the experiments were conducted on a regular workstation (CPU: Intel(R) Core(TM) i7-6850k CPU @ 3.60GHz; GPU: GTX 1080).

The proposed network receives an image as input; at the first stage the default boxes are created for all classes together. Following this, a confidence threshold of 0.01 is applied per class to obtain positive boxes. Finally, non-maximal suppression (NMS) is applied to eliminate the overlapping boxes per class, using a Jaccard Index as the measure of overlap and an overlap threshold of 0.45. NMS essentially does the following (per class): boxes sorted depending on their confidence scores, selecting the box which has the largest confidence score, following which the Jaccard Index is used to remove all the other predicted boxes where Jaccard overlap > the NMS threshold (0.45 here); the previous process is repeated until all boxes are covered.

Two steps were used for training: in the first step a proposed dataset was used to train the network for 35,000 iterations. In the second step, the augmentation tool, which is proposed with the dataset, was used to generate samples to fine-tune the model. The fine-tuning iterations were set to 10,000. Furthermore, the original SSD was trained under the same situation. Figure 6.11 shows the chart of classification loss. It is clear that classification loss dropped from high value gradually as much as the number of iterations increased, which means that more training improved classification.

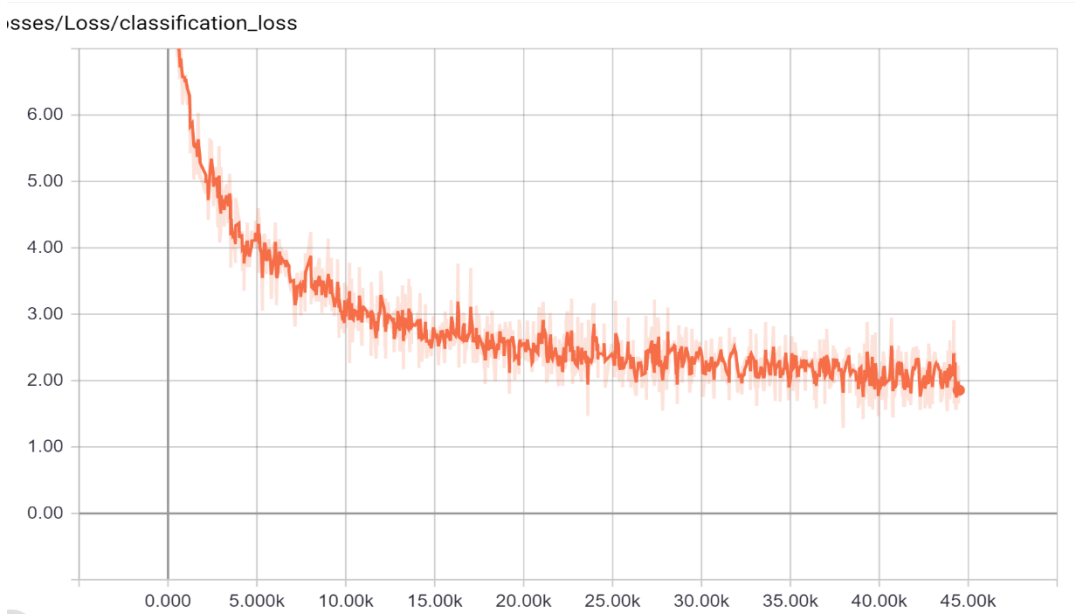


Figure 6.11: Classification performance of the proposed method.

Furthermore, the localisation loss chart was used to study the performance of the proposed method. Figure 6.12 shows that the proposed method reached very good localisation performance when the number of iterations was at almost 45,000.

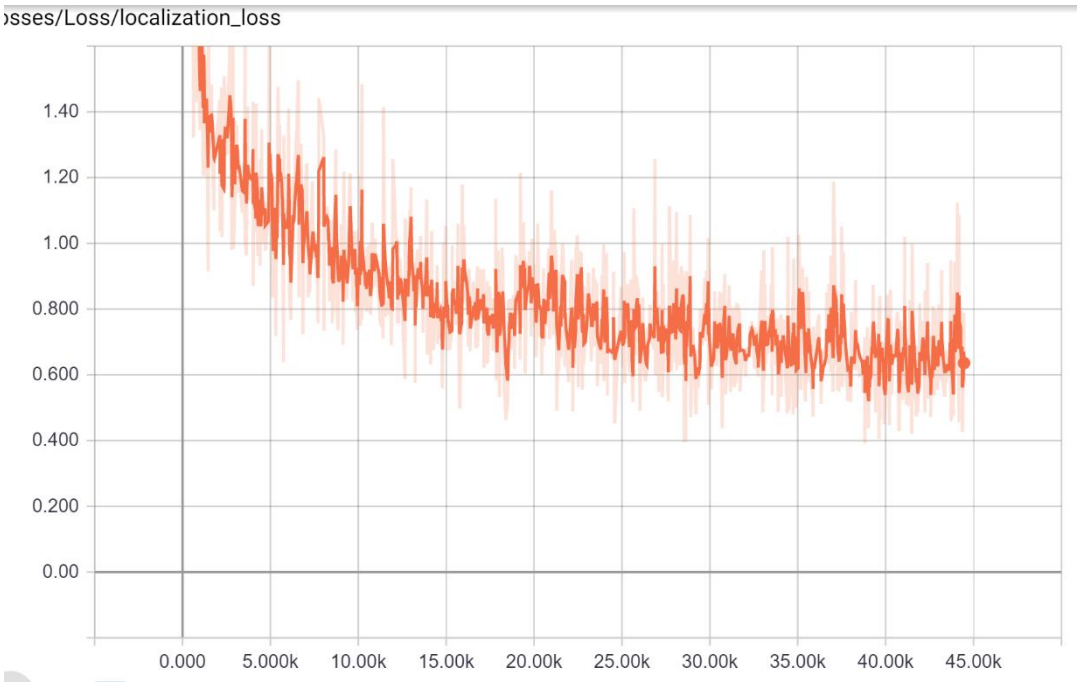


Figure 6.12: The relation between localisation and number of training iterations.

Moreover, the confidence loss decreased gradually to reach a good level at iteration number 45,000. This is shown in Figure 6.13.

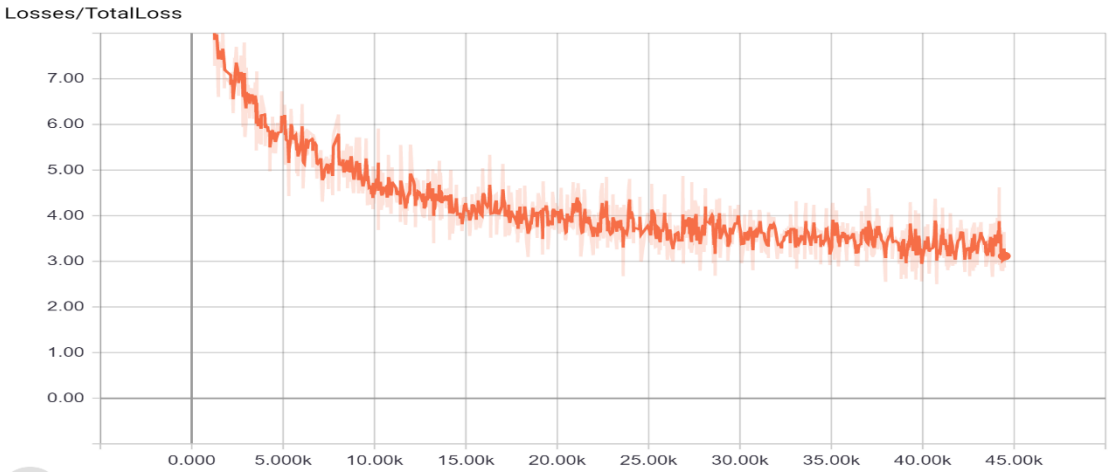


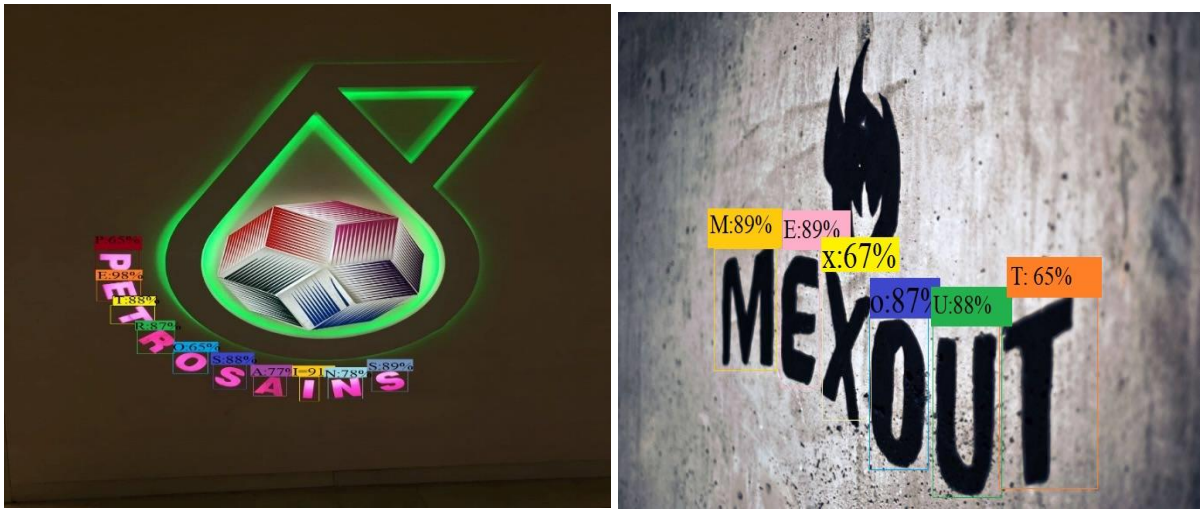
Figure 6.13: The confidence loss of the proposed method.

6.2.9 Experiments

The detection and recognition tasks in the proposed approach were tested on the new proposed datasets, ICDAR2013, ICDAR2015, SVT, and Total-Text, so as to evaluate their text spotting performance. The results are summarised and compared with other methods in Table 6.2. and 6.3, and shown in Figure 6.13. The DetEval protocol (Wolf & Jolion 2006) and F-measure were used to evaluate the results, since the F-measure is the most accurate measurement of detection and recognition of performance.


B: 95%
O: 89%
O: 76%
T: 89%
PRIMARK
SOLE
TRADER
H: 48%
E: 47%
2: 52%
7: 66%
0: 75%
0: 50%
T: 39%

(a)


M:89%
E:89%
X:67%
T: 65%
O:87%
U:88%
MEXOUT

(b)

(c)

Figure 6.14: The results on the proposed dataset (top), Total-Text (middle), and SVT (bottom).

Table 6.2: The performance of the proposed method when applying on the propose dataset

Method	Recall	Precision	F-measure
Proposed method	87.53	93.34	90.34
SSD	69.33	72.45	70.89

Table 6.2 shows that the proposed method performed better than the original SSD; this is due to the added inception layers, which increased the ability of the network to detect small regions. The introduced structure by increasing the type of convolution kernels. Which led to expand the receptive field range helped to increases model's sensitivity to small objects without losing large objects.

The results show that the method also has the ability to detect and recognise digits (see Figure 6.14). To the best of my knowledge, most of the proposed methods have focused on word spotting; for this reason, they missed digits spotting, where the proposed method can spot characters, digits, and words.

Table 6.3 shows the performance of the proposed method and other methods when it comes to text spotting on ICDAR2015, ICDAR2013, and SVT. On ICDAR2013, the method achieved a 91.9% F-measure, which is the second-best accuracy.

Furthermore, the performance of the proposed method was tested on the curve dataset, such as Total-Text. Table 6.4 shows the performance of the method compared with other methods.

The proposed method exploited the idea of individual-character detection and recognition, which made the method more robust and gave it the ability to spot texts, which achieved more robust and superior performance to spot texts of arbitrary shapes. Datasets such as the proposed

dataset and Total-Text have a set of deformations such as arbitrarily-shaped text. Therefore, the evaluation of those datasets is very restricted because of their quadrilateral-based properties.

Table 6.3: Performance of different methods on text Spotting tasks on ICDAR2015, ICDAR2013, SVT (F-measures)

Method	ICDAR2015	ICDAR2013	SVT
(Jaderberg *et al.*, 2016)	-	86.4	53
Gupta (Gupta, Vedaldi and Zisserman, 2016)	-	84.7	55.7
TextBoxs (Liao *et al.*, 2017)	-	91.6	-
(He *et al.*, 2018)	82	91	-
FOTS(X. Liu *et al.*, 2018)	83.55	91.99	-
Maskspotter (Lyu, Liao, *et al.*, 2018)	79.3	92.2	-
Proposed methods	84.5	91.9	54.8

Table 6.4: Performance of different methods on the Total-Text dataset

Method	Recall	Precision	F-measure
MaskSpotter (Lyu, Liao, et al., 2018)	55.0	69.0	61.3
TextSnake (Long *et al.*, 2018)	74.5	82.7	78.4
Proposed method	72.34	72.45	74.79

6.3 Conclusions

In this chapter, a novel framework for horizontal, multi-oriented and curved scene text spotting has been proposed. The method can detect and recognise individual characters and digits. Inception layers are added to the SSD networks and the aspect ratio of the characters is considered, as it is different from that of other objects. Compared with other methods, the proposed method is capable of detecting and recognising characters by end-to-end model completely. The proposed method has the ability to spot text of an arbitrary shape (horizontal, oriented, and curved scene text) with F-measure accuracy = 90.34% for the proposed dataset, and 84.5%, 91.9%, and 54.8% for the ICDAR2015, ICDAR2013, and SVT datasets respectively. The performances on most public datasets show the ability of the method without fine-tuning and the increase in the size of SSD helps to increase the accuracy of spotting. The methods increased the size of the network in depth and width by adding inception layers. The added layers gave the method robustness and flexibility to spot texts of different sizes, because

the added layers had different receptive field sizes. This variety of receptive fields in different layers boosted the performance of the network to spot tests of different sizes.

Chapter 7

Conclusions and Further Works

This thesis presents novel methods in the general areas of object detection and text detection and recognition.

7.1 Conclusions.

The first original contribution made within the context of the research presented in this thesis was an approach to text detection. MSER has been widely used for text detection as it has the ability to extract the characters' regions irrespective of their scale and noise, and to affine the illumination variations (Matas et al., 2004)(Neumann & Matas, 2011). The disadvantage of MSER is that it also detects non-text regions. Two contributions are proposed to overcome the disadvantage of MSER by using classification methods, wherein the text-region candidates extracted by MSER are classified as text and non-text regions. The two main contributions are the use of a combination of a small set of heterogeneous features which are spatially combined to build a large set of features and the selection of the appropriate features to distinguish between the text and the non-text regions. A feature descriptor was calculated using GLCM, LBP, HOG, and aspect ratio. The re-identification performances of the SVM, MLP, and RF classifiers were compared in terms of accuracy. The ICDAR 2003 and ICDAR 2011 datasets were used as a benchmark in our experiments. The results showed that the combination of HOG + LBP + GLCM + AR yielded the best accuracy followed by the combination of LBP + GLCM on the tested datasets. This approach achieved a detection accuracy of 81% (*F*-measure) for ICDAR 2003 and 83% for ICDAR 2011. Moreover, the proposed approach was compared with similar detection methods from the literature, and the results showed that its performance was better than the other methods'.

With a heterogeneous feature set, the combination of feature complexity in the feature selection algorithm supports reducing the overall complexity of the classifier. Furthermore, we took into account the computational load, which is an important consideration in real-time applications. The results showed that using a suitable feature selection and combination approach could significantly increase the accuracy of the algorithms.

This thesis also proposes a new dataset of English text in natural images. This contribution is motivated by the lack of a large number of annotated training images which should include

a range of variation and diversity of texts in natural images for deep learning methods that require massive training data. Moreover, the proposed dataset was annotated on the character level, wherein most of the existing datasets were annotated on the line or word level and there was a lack of character-level annotation datasets. The proposed dataset included 38,500 samples of English characters and 12,500 words in more than 2,100 images. Two annotation methods were used to build the ground truth of the text in the proposed dataset. We obtained character-level and word-level annotations provided in this dataset. Rectangular bounding boxes and polygon shapes with an adaptive number of corner points were used to annotate text in the proposed database.

Furthermore, an augmentation tool has been proposed as a second contribution in Chapter 5, which is provided with the proposed dataset. The proposed augmentation tool has the ability to provide the new location of the bounding boxes after applying transformations on the images. Here, the position of the bounding boxes and the class can be obtained automatically from the original image. This technique helps to increase the number of samples in the dataset and reduce the time of annotations. We proposed two types of augmentations, which are invariant position, where the position of the bounding boxes keep their position such as adding noise, lightning and contrast, and variant position, where the bounding boxes should be located in a new position depending on the augmentation transformations applied to the original image. This tool helps to increase the number of training samples without the need for re-annotating new images created by using the proposed augmentation tool.

Finally, an end-to end detection and recognition approach has been proposed by detecting and recognising each character individually. The proposed framework is an end-to-end character detection and recognition system designed using an improved SSD convolutional neural network, wherein inception layers are added to the SSD networks and the aspect ratio of the characters is considered because it is different from that of the other objects. Compared with the other methods, the proposed method can detect and recognise characters by training an end-to-end model completely. The method achieves an F-measure accuracy of 90.34% for the proposed dataset, and 84.5%, 91.9%, and 54.8% for the ICDAR 2015, ICDAR 2013, and SVT datasets, respectively.

The proposed method has the ability to spot specific text in arbitrarily shaped (horizontal, oriented, and curved) scene text. The addition of inception layers gives the proposed structure the ability to spot text better than the original SSD. The increase in the depth and the width of

the structure improved the performance, as it had various kernel sizes to help to spot text of various sizes and in different directions.

The proposed methods achieved good detection and recognition results in different situations. However, there are still many modifications and extensions that can be applied to improve the robustness of their detection and recognition.

7.2 Future Work

The proposed methods achieved good detection and recognition results in different situations. However, there are still many modifications and extensions that can be applied to improve the robustness of their detection and recognition.

Text recognition can be added to extend the approach proposed in Chapter 4, where the detected regions can be fed to the recognition approach. Furthermore, to improve the work of the method proposed in Chapter 4 and achieve higher accuracy, other combinations of features can be used along with other classifiers such as SIFT features ASIFT, PCA-SIFT, wavelets, and contourlets. These new feature descriptions can be applied to generate more accurate detection and recognition approaches.

The dataset proposed in Chapter 5 can be extended by adding more images and annotating them at the line level. Furthermore, the augmentation tool can be improved by adding more transformations, which will increase the number of training samples.

In Chapter 6, the proposed method has been used to detect and recognise text in images. This framework can be extended for application to videos and GIF files. Furthermore, it can be used to spot text in real-time. Moreover, the performance of the proposed structure can be improved by adding more layers with various kernel sizes. The accuracy of text spotting can be enhanced by increasing the number of training samples and using texts with different orientations for the training.

References

Abadi, M. *et al.* (2016) 'Tensorflow: Large-scale machine learning on heterogeneous distributed systems', *arXiv preprint arXiv:1603.04467*.

Alsadegh, S. S. M. and Lu, J. (2015) 'Analysis of GLCM Parameters for Textures Classification on UMD Database Images', in *The Fifth International Conference on Advanced Communications and Computation.*

Angadi, S. A. and Kodabagi, M. M. M. (2009) 'A Texture Based Methodology for Text Region Extraction from Low Resolution Natural Scene Images', *International Journal of Image Processing (IJIP)*, 3(5), pp. 229–245.

Asperti, A. and Mastronardo, C. (2017) 'The effectiveness of data augmentation for detection of gastrointestinal diseases from endoscopical images', *arXiv preprint arXiv:1712.03689*.

Baek, Y. *et al.* (2019) 'Character Region Awareness for Text Detection', pp. 9365–9374. Available at: http://arxiv.org/abs/1904.01941.

Bai, B., Yin, F. and Liu, C. L. (2012) 'A fast stroke-based method for text detection in video', *Proceedings - 10th IAPR International Workshop on Document Analysis Systems, DAS 2012*, pp. 69–73. doi: 10.1109/DAS.2012.3.

Bissacco, A. Cummins, M., Netzer, Y., Neven, H., (2013) 'PhotoOCR: Reading text in uncontrolled conditions', in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 785–792. doi: 10.1109/ICCV.2013.102.

Bloice, M. D., Stocker, C. and Holzinger, A. (2017) 'Augmentor: an image augmentation library for machine learning', *arXiv preprint arXiv:1708.04680*.

Bosch, A., Zisserman, A. and Mu, X. (2007) 'IEEE 2007 - Image Classification using Random Forests and Ferns.pdf', *2007 IEEE 11th international conference on computer vision*, pp. 1--8.

Breiman, L. (2001) 'Random forests', *Machine learning*. Springer, 45(1), pp. 5–32.

Burges, C. J. C. (1998) 'A Tutorial on Support Vector Machines for Pattern Recognition', *Data Mining and Knowledge Discovery*, 2, pp. 121–167. doi: 10.1023/A:1009715923555.

Bušta, M., Neumann, L. and Matas, J. (2017) 'Deep textspotter: An end-to-end trainable scene text localization and recognition framework', in *Computer Vision (ICCV), 2017 IEEE International Conference on*, pp. 2223–2231.

Ch'Ng, C. K. and Chan, C. S. (2018) 'Total-Text: A Comprehensive Dataset for Scene Text Detection and Recognition', *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 1, pp. 935–942. doi: 10.1109/ICDAR.2017.157.

Chaddad, A. Ahmad, F., Amin, M. G., Sevigny, P., DiFilippo, D. ,(2014) 'Textural feature selection for enhanced detection of stationary humans in through-the-wall radar imagery', *Radar Sensor Technology XVIII*, 9077(May 2014), p. 90770F. doi: 10.1117/12.2049416.

Chen, H. Tsai, S., Schroth, G., Chen, D., Grzeszczuk, R., Girod, B.,(2011) 'Robust text detection in natural images with edge-enhanced maximally stable extremal regions',

Proceedings - International Conference on Image Processing, ICIP, pp. 2609–2612. doi: 10.1109/ICIP.2011.6116200.

Chen, X. and Yuille, A. L. (2004) 'Detecting and Reading Text in Natural Scenes', in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)*, pp. 366–373.

Cheng, Z., Bai, F., Xu, Y., Zheng, G., Pu, S. and Zhou, S., (2017). Focusing attention: Towards accurate text recognition in natural images. In *Proceedings of the IEEE international conference on computer vision* (pp. 5076-5084).

Chollet, F. (2015) *Keras: Deep Learning for humans*, *Github*. Available at: https://github.com/keras-team/keras.

Chowdhury, A. R., Bhattacharya, U. and Parui, S. K. (2012) 'Scene text detection using sparse stroke information and MLP.', in *21st International Conference on Pattern Recognition (ICPR 2012)*, pp. 294–297. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6460130.

Clausi, D. A. (2002) 'An analysis of co-occurrence texture statistics as a function of grey level quantization', *Canadian Journal of Remote Sensing*. Taylor & Francis, 28(1), pp. 45–62. doi: 10.5589/m02-004.

Collobert, R., Bengio, S. and Mariéthoz, J. (2002) *Torch: a modular machine learning software library*.

Dalal, N. and Triggs, B. (2005) 'Histograms of oriented gradients for human detection', in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, pp. 886–893.

Deng, D. Liu, H., Li, X., Cai, D.,(2018) 'PixelLink: Detecting Scene Text via Instance Segmentation', *arXiv preprint arXiv:1801.01315*.

Dollár, P. and Zitnick, C. L. (2015) 'Fast edge detection using structured forests', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(8), pp. 1558–1570. doi: 10.1109/TPAMI.2014.2377715.

Epshtein, B., Ofek, E. and Wexler, Y. (2010) 'Detecting text in natural scenes with stroke width transform', in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2963–2970.

Erhan, D. Szegedy, C., Toshev, A., Anguelov, D., (2014) 'Scalable object detection using deep neural networks', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2147–2154.

Feild, J. L. and Learned-Miller, E. G. (2013) 'Improving open-vocabulary scene text recognition', in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*. IEEE, pp. 604–608. doi: 10.1109/ICDAR.2013.125.

Gatos, B. Pratikakis, I., Kepene, K., Perantonis, S J., (2005) 'Text detection in indoor/outdoor scene images', in *Proc. First Workshop of Camera-based Document Analysis and Recognition*, pp. 127–132. Available at: http://iit.demokritos.gr/~bgat/cbdar2005.pdf.

George, M. and Zwiggelaar, R. (2019) 'Comparative Study on Local Binary Patterns for Mammographic Density and Risk Scoring', *Journal of Imaging*, 5(2), p. 24. doi: 10.3390/jimaging5020024.

Girshick, R. (2015) 'Fast r-cnn', in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448.

Gllavata, J., Ewerth, R. and Freisleben, B. (2004) 'Text Detection in Images Based on Unsupervised Classification of High- Frequency Wavelet Coefficients', *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, 1, pp. 2–5.

Goel, V. Mishra, A., Alahari, K., Jawahar, C. V., (2013) 'Whole is greater than sum of parts: Recognizing scene text words', *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*. IEEE, pp. 398–402. doi: 10.1109/ICDAR.2013.87.

Gupta, A., Vedaldi, A. and Zisserman, A. (2016) 'Synthetic data for text localisation in natural images', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2315–2324.

Gupta, N. and Banga, V. K. (2012) 'Localization of Text in Complex Images Using Haar Wavelet Transform', *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 1(6), pp. 111–115.

Hall, M. A. and Smith, L. A. (1999) 'Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper.', in *FLAIRS conference*, pp. 235–239.

Haralick, R. M., Shanmugam, K. and others (1973) 'Textural features for image classification', *IEEE Transactions on systems, man, and cybernetics*. Ieee, (6), pp. 610–621.

He, D. Yang, X., Liang, C., Zhou, Z., Ororbi, A., Kifer, D., Lee Giles, C., (2017) 'Multi-scale FCN with cascaded instance aware segmentation for arbitrary oriented word spotting in the wild', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3519–3528.

He, P. Huang, W., Qiao, Y., Loy, C., Tang, X., (2016) 'Reading Scene Text in Deep Convolutional Sequences.', in *AAAI*, pp. 3501–3508.

He, P. Huang, W., He, T., Zhu, Q., Qiao, Y., Li, X., (2017) 'Single shot text detector with regional attention', in *The IEEE International Conference on Computer Vision (ICCV)*.

He, T. Tian, Z., Huang, W., Shen, C., Qiao, Y., Sun, C., Technologies, M., (2018) 'An end-to-end TextSpotter with Explicit Alignment and Attention', in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

He, W. Zhang, X., Yin, F., Liu, C., (2017) 'Deep direct regression for multi-oriented scene text detection', *In Proceedings of the IEEE International Conference on Computer Vision*, pp. 745–753.

Hu, H., Zhang, C., Luo, Y., Wang, Y., Han, J. and Ding, E., (2017). *Wordsup: Exploiting word annotations for character based text detection. In Proceedings of the IEEE International*

Conference on Computer Vision, pp. 4940-4949.

Huang, D. Shan, C., Ardabilian, M., Wang, Y., Chen, L., (2011) 'Local binary patterns and its application to facial image analysis: a survey', *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*. IEEE, 41(6), pp. 765–781.

Huang, R., Shivakumara, P. and Uchida, S. (2013) 'Scene character detection by an edge-ray filter', in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pp. 462–466. doi: 10.1109/ICDAR.2013.99.

Huang, W. Lin, Z., Yang, J., Wang, J., (2013) 'Text Localization in Natural Images using Stroke Feature Transform and Text Covariance Descriptors', *Proceedings of the 2013 IEEE International Conference on Computer Vision*, pp. 1241--1248.

Huang, W., Qiao, Y. and Tang, X. (2014) 'Robust Scene Text Detection with Convolution Neural Network Induced MSER Trees', *Computer Vision – ECCV 2014*, 8692, pp. 497–511.

Jaderberg, M. Simonyan, K., Vedaldi, A., Zisserman, A., (2014) 'Synthetic data and artificial neural networks for natural scene text recognition', *arXiv preprint arXiv:1406.2227*.

Jaderberg, M. Simonyan, K.,Vedaldi, A., Zisserman, A., (2016) 'Reading text in the wild with convolutional neural networks', *International Journal of Computer Vision*. Springer, 116(1), pp. 1–20.

Jaderberg, M., Vedaldi, A. and Zisserman, A. (2014) 'Deep features for text spotting', in *European conference on computer vision*, pp. 512–528. doi: 10.1007/978-3-319-10593-2_34.

Jiang, Y. Zhu, X., Wang, X., Yang, S., Li, W., Wang, H., Fu, P., Luo, Z.,(2017) 'R2CNN: rotational region CNN for orientation robust scene text detection', *arXiv preprint arXiv:1706.09579*.

Jung, A. (2017) 'imgaug: Image augmentation for machine learning experiments.', p. Accessed 3 April 2017. Available at: https://github.com/aleju/imgaug.

Jung, K., Kim, K. I. and Jain, A. K. (2004) 'Text information extraction in images and video: a survey', *Pattern recognition*. Elsevier, 37(5), pp. 977–997. doi: 10.1016/j.patcog.2003.10.012.

Karatzas, D. Mestre, S., Mas, J., Nourbakhsh, F., Roy, P., (2011) 'ICDAR 2011 robust reading competition-challenge 1: reading text in born-digital images (web and email)', in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pp. 1485–1490.

Karatzas, D. Shafait, F., Uchida, S., Iwamura, M., Gomez, L., Mestre, S., Mas, J., Mota, D.,Almaz, J., (2013) 'ICDAR 2013 Robust Reading Competition', in *ICDAR 2013 robust reading competition.* doi: 10.1109/ICDAR.2013.221.

Karatzas, D. Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Luka, N., Vijay R., Lu, S., *et al.* (2015) 'ICDAR 2015 competition on robust reading', in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pp. 1156–1160.

Karatzas, D. and Antonacopoulos, A. (2004) 'Text extraction from web images based on a split-and-merge segmentation method using colour perception', in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, pp. 634–637.

Kim, H.-K. (1996) 'Efficient Automatic Text Location Method and Content-Based Indexing and Structuring of Video Database.pdf', *Journal of Visual Communication and Image Representation*, 7(4), pp. 336–344. doi: 10.1006/jvci.1996.0029.

Kim, K. I., Jung, K. and Kim, J. H. (2003) 'Texture-Based Approach for Text Detection in Images Using Support Vector Machines and Continuously Adaptive Mean Shift Algorithm æ', *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 25(12), pp. 1631–1639.

Ko, B. C., Kim, S. H. and Nam, J. Y. (2011) 'X-ray image classification using random forests with local wavelet-based CS-local binary patterns', *Journal of Digital Imaging*, 24(6), pp. 1141–1151. doi: 10.1007/s10278-011-9380-3.

Koo, H. Il and Kim, D. H. (2013) 'Scene Text Detection via Connected Component Clustering and Nontext Filtering', *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 22(6), pp. 2296–2305.

Lam, O. Dayoub, F., Schulz, R., Corke, P., (2014) 'Text recognition approaches for indoor robotics : a comparison', *2014 Australasian Conference on Robotics and Automation*, (December), pp. 2–4.

Lee, C.-Y. and Osindero, S. (2016) 'Recursive recurrent nets with attention modeling for ocr in the wild', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2231–2239.

Lee, C. Y. Bhardwaj, A., Di, W., Jagadeesh, V., Piramuthu, R., (2014) 'Region-based discriminative feature pooling for scene text recognition', *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4050–4057. doi: 10.1109/CVPR.2014.516.

Lee, J.-J. Lee, P., Lee, S.,Yuille, A., Koch, C.,(2011) 'Adaboost for text detection in natural scene', in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pp. 429–434.

Li, H., Wang, P. and Shen, C. (2017) 'Towards end-to-end text spotting with convolutional recurrent neural networks', in *Proc. ICCV*, pp. 5238–5246.

Liang, J., Doermann, D. and Li, H. (2005) 'Camera-based analysis of text and documents : a survey', *International Journal on Document Analysis and Recognition*, 7, pp. 84–104.

Liao, M. Shi, B., Bai, X., Wang, X., Liu, W.,(2017) 'TextBoxes: A Fast Text Detector with a Single Deep Neural Network.', in *AAAI*, pp. 4161–4167.

Lienhart, R. and Wernicke, A. (2002) 'Localizing and segmenting text in images and videos', *IEEE Transactions on Circuits and Systems for Video Technology*, 12(4), pp. 256–268. doi: 10.1109/76.999203.

Liu, J. Li, H., Zhang, S., Liang, W., (2011) 'A Novel Italic Detection and Rectification Method for Chinese Advertising Images', *2011 International Conference on Document Analysis and Recognition*, pp. 698–702. doi: 10.1109/ICDAR.2011.146.

Liu, W. Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., Berg, A., (2016) 'Ssd: Single shot multibox detector', in *European conference on computer vision*, pp. 21–37.

Liu, X. Liang, D., Yan, S., Chen, D., Qiao, Y., Yan, J.,(2018) 'FOTS : Fast Oriented Text Spotting with a Unified Network', in *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5676–5685.

Liu, X., Meng, G. and Pan, C. (2019) 'Scene text detection and recognition with advances in deep learning: a survey', *International Journal on Document Analysis and Recognition (IJDAR)*. Springer Berlin Heidelberg, pp. 1–20. doi: 10.1007/s10032-019-00320-5.

Liu, Y. and Jin, L. (2017) 'Deep matching prior network: Toward tighter multi-oriented text detection', in *Proc. CVPR*, pp. 3454–3461.

Liu, Z. Li, Y., Ren, F., Goh, W.L., Yu, H., (2018) 'Squeezedtext: A real-time scene text recognition by binary convolutional encoder-decoder network', in *Thirty-Second AAAI Conference on Artificial Intelligence*.

Long, S. Ruan, J., Zhang, W., He, X., Wu, W., Yao, C., (2018) 'TextSnake: A Flexible Representation for Detecting Text of Arbitrary Shapes', in *Lecture Notes in Computer Science*, pp. 19–35. doi: 10.1007/978-3-030-01216-8_2.

Long, S., He, X. and Ya, C. (2018) 'Scene Text Detection and Recognition: The Deep Learning Era', *arXiv preprint arXiv:1811.04256*.

Lou, X. Boukharouba, K., Boonært, J., Fleury, A., Lecoeuche, S., (2016) 'Generative shape models: Joint text recognition and segmentation with very little training data', in *Advances in Neural Information Processing Systems*, pp. 2793–2801.

Lu, Y. *et al.* (2014) 'Application of an incremental SVM algorithm for on-line human recognition from video surveillance using texture and color features', *Neurocomputing*. Elsevier, 126, pp. 132–140.

Lucas, S. M. Panaretos, A., Sosa, L., Tang, A., Wong, S. and Young, R., (2003) 'ICDAR 2003 Robust Reading Competitions', in *INTERNATIONAL CONFERENCE ON DOCUMENT ANALYSIS AND RECOGNITION (ICDAR)*.

Lucas, S. M. Panaretos, A., Sosa, L., Tang, A., Wong, S., Young, R., Ashida, K., Nagai, H., Okamoto, M., Yamamoto, H. and Miyao, H., (2005) 'ICDAR 2003 robust reading competitions: entries, results, and future directions', *International Journal of Document Analysis and Recognition (IJDAR)*. Springer, 7(2–3), pp. 105–122.

Lucas, S. M. (2005) 'ICDAR 2005 Text Locating Competition Results', in *INTERNATIONAL CONFERENCE ON DOCUMENT ANALYSIS AND RECOGNITION (ICDAR), 2005*, pp. 0–4.

Luo, W., Li, Y., Urtasun, R., Zemel, R., (2017) 'Understanding the Effective Receptive Field in Deep Convolutional Neural Networks', *CoRR*, abs/1701.0. Available at:

http://arxiv.org/abs/1701.04128.

Lyu, P., Liao, M., *et al.* (2018) 'Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes', in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 67–83.

Lyu, P., Yao, C.,Wu, W., Yan, S., Bai, X., (2018) 'Multi-oriented scene text detection via corner localization and region segmentation', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7553–7563. doi: 10.1109/CVPR.2018.00788.

Matas, J. Chum, O., Urban, M., Pajdla, T., (2004) 'Robust wide-baseline stereo from maximally stable extremal regions', *Image and vision computing*. Elsevier, 22(10), pp. 761–767.

Mikołajczyk, A. and Grochowski, M. (2018) 'Data augmentation for improving deep learning in image classification problem', *2018 International Interdisciplinary PhD Workshop, IIPhDW 2018*. IEEE, pp. 117–122. doi: 10.1109/IIPHDW.2018.8388338.

Mishra, A., Alahari, K. and Jawahar, C. (2012) 'Scene Text Recognition using Higher Order Language Priors', *Procedings of the British Machine Vision Conference 2012*, pp. 127.1-127.11. doi: 10.5244/C.26.127.

Mishra, A., Alahari, K. and Jawahar, C. V. (2012) 'Top-down and bottom-up cues for scene text recognition', *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 2687–2694. doi: 10.1109/CVPR.2012.6247990.

Mosleh, A., Bouguila, N. and Hamza, a Ben (2012) 'Image Text Detection Using a Bandlet-Based Edge Detector and Stroke Width Transform', *Proceedings of the British Machine Vision Conference 2013*, pp. 1–12. doi: 10.5244/C.26.63.

Nakajima, C. Pontil, M., Heisele, B., Poggio, T., (2003) 'Full-body person recognition system', *Pattern Recognition*, 36(9), pp. 1997–2006. doi: 10.1016/S0031-3203(03)00061-X.

Nayef, N. Y., Bizid, I., Choi, H., Feng, Y., Karatzas, D., Luo, Z., Pal, U., Rigaud, C., Chazalon, J., Khlif, W., Luqman, M., Burie, J., Liu, C.,Ogie, J., (2018) 'ICDAR2017 Robust Reading Challenge on Multi-Lingual Scene Text Detection and Script Identification - RRC-MLT', *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 1, pp. 1454–1459. doi: 10.1109/ICDAR.2017.237.

Netzer, Y. and Wang, T. (2011) 'Reading digits in natural images with unsupervised feature learning', *In NIPS workshop on deep learning and unsupervised feature learning*, 5, pp. 1–9. doi: 10.2118/18761-MS.

Neumann, L. and Matas, J. (2011) 'A method for text localization and recognition in real-world images', in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 770–783. doi: 10.1007/978-3-642-19318-7_60.

Neumann, L. and Matas, J. (2012) 'Real-time scene text localization and recognition', in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3538–3545. doi: 10.1109/CVPR.2012.6248097.

Neumann, L. and Matas, J. (2013) 'On combining multiple segmentations in scene text recognition', *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, pp. 523–527. doi: 10.1109/ICDAR.2013.110.

Ning, C. Zhou, H., Song, Y., Tang, J., (2017) 'Inception Single Shot MultiBox Detector for object detection', *2017 IEEE International Conference on Multimedia and Expo Workshops, ICMEW 2017*, (July), pp. 549–554. doi: 10.1109/ICMEW.2017.8026312.

Novikova, T. Barinova, O., Kohli, P., Lempitsky, V., (2012) 'Large-Lexicon Attribute-Consistent Text Recognition in Natural Images', in Fitzgibbon, A. et al. (eds) Computer Vision -- ECCV 2012. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 752–765.

Pan, Y., Hou, X. and Liu, C. (2008) 'A Robust System to Detect and Localize Texts in Natural Scene Images', *The Eighth IAPR Workshop on Document Analysis Systems*, pp. 35–42. doi: 10.1109/DAS.2008.42.

Pan, Y., Hou, X. and Liu, C. (2011) 'A Hybrid Approach to Detect and Localize Texts in Natural Scene Images', *IEEE Transactions on Image Processing*. IEEE, 20(3), pp. 800–813. doi: 10.1109/tip.2010.2070803.

Park, J. Lee, G., Kim, E., Lim, J.,Kim, S.,Yang, H., Lee, M., Hwang, S., (2010) 'Automatic detection and recognition of Korean text in outdoor signboard images', *Pattern Recognition Letters*. Elsevier B.V., 31(12), pp. 1728–1739. doi: 10.1016/j.patrec.2010.05.024.

Phan, T. Q. Shivakumara, P., Tian,S., Tan, C., (2013) 'Recognizing Text with Perspective Distortion in Natural Scenes', in *2013 IEEE International Conference on Computer Vision*, pp. 569–576. doi: 10.1109/ICCV.2013.76.

Pietikäinen, M. Hadid A., Zhao, G., Ahonen, T., (2013) Computer Vision Using Local Binary Patterns. Springer London (Computational Imaging and Vision). Available at: https://books.google.co.uk/books?id=vDBAngEACAAJ.

Risnumawan, A. Shivakumara, P., Chan, C., Tan, C., (2014) 'A robust arbitrary text detection system for natural scene images', Expert Systems with Applications. Elsevier Ltd, 41(18), pp. 8027–8048. doi: 10.1016/j.eswa.2014.07.008.

Schneider, Philip K. Eberly, D. H. (2003) Geometric Tools for C*omputer Graphics*. Morgan Kaufmann. p. 98. ISBN 978-1-55860-594-7.

Seeri, S. V, Pujari, J. D. and Hiremath, P. S. (2015) 'Multilingual text localization in natural scene images using wavelet based edge features and fuzzy classification', *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, 4(1), pp. 210–218.

Sermanet, P. Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y., (2013) 'Overfeat: Integrated recognition, localization and detection using convolutional networks', *arXiv preprint arXiv:1312.6229*.

Shahab, A., Shafait, F. and Dengel, A. (2011) 'ICDAR 2011 Robust Reading Competition Challenge 2 : Reading Text in Scene Images', in *INTERNATIONAL CONFERENCE ON DOCUMENT ANALYSIS AND RECOGNITION (ICDAR)*. doi: 10.1109/ICDAR.2011.296.

Shalev-Shwartz, S. and Ben-David, S. (2013) *Understanding machine learning: From theory to algorithms*, *Understanding Machine Learning: From Theory to Algorithms*. doi: 10.1017/CBO9781107298019.

Shi, B. Wang, X., Lyu, P., Yao, C., Bai, X., (2016) 'Robust scene text recognition with automatic rectification', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4168–4176.

Shi, B. Yang, M., Wang, X., Lyu, P., Yao, C., Bai, X., (2018) 'Aster: An attentional scene text recognizer with flexible rectification', *IEEE transactions on pattern analysis and machine intelligence*. IEEE.

Shi, B., Bai, X. and Belongie, S. (2017) 'Detecting oriented text in natural images by linking segments', *arXiv preprint arXiv:1703.06520*.

Shi, B., Bai, X. and Yao, C. (2017) 'An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition', *IEEE transactions on pattern analysis and machine intelligence*. IEEE, 39(11), pp. 2298–2304.

Shi, C., Wang, C., Xiao, B., Zhang, Y. and Gao, S. (2013) 'Scene text detection using graph model built upon maximally stable extremal regions', *Pattern Recognition Letters*. Elsevier B.V., 34(2), pp. 107–116. doi: 10.1016/j.patrec.2012.09.019.

Shi, C., Wang, C., Xiao, B., Zhang, Y., Gao, S., *et al.* (2013) 'Scene text recognition using part-based tree-structured character detection', *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2961–2968. doi: 10.1109/CVPR.2013.381.

Shivakumara, P., Huang, W. and Tan, C. L. (2008) 'An Efficient Edge based Technique for Text Detection in Video Frames', *The Eighth IAPR Workshop on Document Analysis Systems*, pp. 307–314. doi: 10.1109/DAS.2008.17.

Shivakumara, P., Phan, T. Q. and Tan, C. L. (2011) 'A Laplacian Approach to Multi-Oriented Text Detection in Video', *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 33(2), pp. 412–419.

Simard, P. Y., Steinkraus, D. and Platt, J. C. (2003) 'Best practices for convolutional neural networks applied to visual document analysis', in *null*, p. 958.

Smith, R. (2011) 'Limits on the Application of Frequency-based Language Models to OCR', in *ICDAR*, pp. 538–542.

Smith, R., Antonova, D. and Lee, D.-S. (2009) 'Adapting the Tesseract open source OCR engine for multilingual OCR', in *Proceedings of the International Workshop on Multilingual OCR*, p. 1.

Song, G. and Tang, S. (1997) 'Method for spectral pattern recognition of color camouflage', *Optical Engineering*. International Society for Optics and Photonics, 36(6), pp. 1779–1782.

Strouthopoulos, C., Papamarkos, N. and Atsalakis, a. E. (2002) 'Text extraction in complex color documents', *Pattern Recognition*, 35(8), pp. 1743–1758. doi: 10.1016/S0031-3203(01)00167-4.

Sun, L. Huo, Q., Jia, W.,Chen, K., (2014) 'Robust text detection in natural scene images by generalized color-enhanced contrasting extremal region and neural networks', in *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pp. 2715–2720.

Sun, L. Huo, Q., Jia, W., Chen, K., (2015) 'A robust approach for text detection from natural scene images', *Pattern Recognition*. Elsevier, 48(9), pp. 2906–2920. doi: 10.1016/j.patcog.2015.04.002.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. (2016) 'Rethinking the inception architecture for computer vision. (pp. 2818-2826).', *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826.

Szegedy, C. Reed, S., Erhan, D., Anguelov, D., Ioffe, S., (2014) 'Scalable, high-quality object detection', *arXiv preprint arXiv:1412.1441*.

Szegedy, C. Liu, W., Jia, Y., Sermanet, Pi.,Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.,(2015) 'Going deeper with convolutions', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9.

Valle, E., Fornaciali, M., Menegola, A., Tavares, J., Bittencourt, F.V., Li, L.T. and Avila, S., 2017. Data, depth, and design: learning reliable models for melanoma screening. *arXiv preprint arXiv:1711.00441*, *1*..

Voisin, A. Krylov, V., Moser, G., Serpico, S., Zerubia, J., (2013) 'Classification of very high resolution SAR images of urban areas using copulas and texture in a hierarchical Markov random field model', *IEEE Geoscience and Remote Sensing Letters*, 10(1), pp. 96–100. doi: 10.1109/LGRS.2012.2193869.

Wang, K., Babenko, B. and Belongie, S. (2011) 'End-to-end scene text recognition', in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1457–1464. doi: 10.1109/ICCV.2011.6126402.

Wang, K. and Belongie, S. (2012) 'Word Spotting in the Wild.pdf', *11th European Conference on Computer Vision*, pp. 591–604.

Wang, X. Song, Y., Zhang, Y., Xin, J., (2015) 'Natural scene text detection with multi-layer segmentation and higher order conditional random field based analysis', *Pattern Recognition Letters*. Elsevier, 60, pp. 41–47. doi: 10.1016/j.patrec.2015.04.005.

Wang, X., Wang, K. and Lian, S. (2019) 'A Survey on Face Data Augmentation', *CoRR*, abs/1904.1. Available at: http://arxiv.org/abs/1904.11685.

Weinman, J. J. Butler, Z., Knoll, D., Feild, J., (2014) 'Toward integrated scene text reading', *IEEE transactions on pattern analysis and machine intelligence*. IEEE, 36(2), pp. 375–387.

Weinman, J. J., Learned-miller, E. and Hanson, A. R. (2009) 'Scene Text Recognition using Similarity and a Lexicon with Sparse Belief Propagation', *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 31(10), pp. 1733–1746.

Wolf, C. and Jolion, J.-M. (2006) 'Object count/area graphs for the evaluation of object detection and segmentation algorithms', *International Journal of Document Analysis and*

Recognition (IJDAR). Springer, 8(4), pp. 280–296.

Wong, S. C. Gatt A., Stamatescu ,V., (2016) 'Understanding Data Augmentation for Classification: When to Warp?', *2016 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2016*. doi: 10.1109/DICTA.2016.7797091.

Wu, V., Manmatha, R. and Riseman, E. M. (1997) 'Finding text in images', in *ACM DL*, pp. 3–12.

Yao, C. Bai, X., Liu, W., Ma, Y., Tu, Z.,(2012) 'Detecting Texts of Arbitrary Orientations in Natural Images', in *IEEE Conf. CVPR*, pp. 1083–1090.

Yao, C., Zhang, X., Bai, X., Liu, W., Ma, Y., Tu, Z.,(2013) 'Rotation-invariant features for multi-oriented text detection in natural images', *PloS one*. Public Library of Science, 8(8), p. e70173. doi: 10.1371/journal.pone.0070173.

Yao, C. Bai, X., Shi, B., Liu, W., (2014) 'Strokelets: A learned multi-scale representation for scene text recognition', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4042–4049. doi: 10.1109/CVPR.2014.515.

Yao, C., Bai, X. and Liu, W. (2014) 'A Unified Framework for Multioriented Text Detection and Recognition', *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 23(11), pp. 4737–4749.

Ye, Q. and Doermann, D. (2013) 'Scene Text Detection via Integrated Discrimination of Component Appearance and Consensus', *in Proc. Int. Workshop Camera-Based Doc. Anal. Recognit.*, pp. 47–59.

Ye, Q. and Doermann, D. (2015) 'Text detection and recognition in imagery: A survey', *IEEE transactions on pattern analysis and machine intelligence*. IEEE, 37(7), pp. 1480–1500.

Yi, C. and Tian, Y. (2011) 'Text string detection from natural scenes by structure-based partition and grouping', *IEEE Transactions on Image Processing*. IEEE, 20(9), pp. 2594–2605.

Yi, C. and Tian, Y. (2012) 'Localizing text in scene images by boundary clustering, stroke segmentation, and string fragment classification', *IEEE Transactions on Image Processing*. IEEE, 21(9), pp. 4256–4268.

Yin, X. X.-C., Huang, K. and Hao, H.-W. (2013) 'Robust Text Detection in Natural Scene Images.', *IEEE transactions on pattern analysis and machine intelligence*, 36(5), pp. 970–983. doi: AE9E97E4-72D3-400F-9AB2-205D825497F4.

Yu, L. and Liu, H. (2003) 'Feature selection for high-dimensional data: A fast correlation-based filter solution', in *Proceedings of the 20th international conference on machine learning (ICML-03)*, pp. 856–863.

Yuliang, L. Lianwen, J., Shuaitao, Z., Sheng, Z., (2017) 'Detecting Curve Text in the Wild: New Dataset and New Solution', *arXiv preprint arXiv:1712.02170*.

Zhang, H., Gao, W., Chen, X. and Zhao, D., (2006). Object detection using spatial histogram

features. *Image and Vision Computing*, *24*(4), pp.327-341.

Zhang, H. Liu, C., Yang, C., Ding, X., Wang, K., (2011) 'An improved scene text extraction method using Conditional Random Field and Optical Character Recognition', *in: 2011 International Conference on Document Analysis and Recognition (ICDAR), IEEE,* (2), pp. 708–712. doi: 10.1109/ICDAR.2011.148.

Zhang, H. Zhao, K., Song, Y., Guo, J., (2013) 'Text extraction from natural scene image: A survey', *Neurocomputing*. Elsevier, 122, pp. 310–323. doi: 10.1016/j.neucom.2013.05.037.

Zhang, Z. Zhang, C., Shen, W., Yao, C., Liu, W., Bai, X.,(2016) 'Multi-Oriented Text Detection With Fully Convolutional Networks', in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Zhao, Y., Lu, T. and Liao, W. (2011) 'A robust color-independent text detection method from complex videos', *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, pp. 374–378. doi: 10.1109/ICDAR.2011.83.

Zhong, Y. and Jain, A. K. (2000) 'Object localization using color, texture and shape', *Pattern Recognition*, 33(4), pp. 671–684. doi: 10.1016/S0031-3203(99)00079-5.

Zhou, X. Zhou, S., Yao, C., Cao, Z., Yin, Q., (2015) 'ICDAR 2015 Text Reading in the Wild Competition', *arXiv:1506.03184 [cs.CV]*.

Zhou, X. Ning, C., Zhou, H., Song, Y., Tang, J., (2017) 'EAST: an efficient and accurate scene text detector', in *Proc. CVPR*, pp. 2642–2651.

Zhu, S. (2015) 'An End-to-End License Plate Localization and Recognition System An End-to-End License Plate Localization and Recognition System'.

Zhu, Y., Yao, C. and Bai, X. (2016) 'Scene text detection and recognition: Recent advances and future trends', *Frontiers of Computer Science*. Springer, 10(1), pp. 19–36.

Zhuo, F. Q., Lin, P. Q. and Gu, Y. M. (2014) 'Vision-Based Vehicle Detection in Real Traffic Environment Using Fast Wavelet Transform and Kalman Filter', *Advanced Materials Research*, 998–999, pp. 717–722. doi: 10.4028/www.scientific.net/amr.998-999.717.

Zitnick, C. L. and Dollár, P. (2014) 'LNCS 8693 - Edge Boxes: Locating Object Proposals from Edges', *Eccv*, pp. 391–405. Available at: https://link.springer.com/content/pdf/10.1007%2F978-3-319-10602-1_26.pdf.