T.R
SAKARYA UNIVERSITY
INSTITUTE OF NATURAL SCIENCE

NATURAL SCENE IMAGE TEXT DETECTION AND RECOGNITION USING A NOVEL GLOBAL CURVATURE FEATURE

Ph.D. THESIS

Belaynesh CHEKOL

Department	**:**	**COMPUTER ENGINEERING**
Field of Science	**:**	**COMPUTER AND INFORMATION ENGINEERING**
Supervisor	**:**	**Assoc. Prof. Dr. Numan ÇELEBİ**

February 2020

T.R
SAKARYA UNIVERSITY
INSTITUTE OF NATURAL SCIENCE

NATURAL SCENE IMAGE TEXT DETECTION AND RECOGNITION USING A NOVEL GLOBAL CURVATURE FEATURE

Ph.D. THESIS

Belaynesh CHEKOL

Department	:	COMPUTER ENGINEERING
Field of Science	:	COMPUTER AND INFORMATION ENGINEERING
Supervisor	:	Assoc. Prof. Dr. Numan ÇELEBİ

This thesis has been accepted unanimously / with majority of votes by the examination committee on 28.02.2020

Prof. Dr.
Nedim TUTKUN
Head of Jury

Prof. Dr.
Raşit KÖKER
Jury Member

Doç. Dr.
Numan ÇELEBİ
Jury Member

n. Çelebi

Doç. Dr.
Devrim AKGÜN
Jury Member

Dr. Öğr. Üyesi
Tuğrul TAŞCI
Jury Member

DECLARATION

I declare that all the data in this thesis was obtained by myself in academic rules, all visual and written information and results were presented in accordance with academic and ethical rules, there is no distortion in the presented data, in case of utilizing other people's works they were refereed properly to scientific norms, the data presented in this thesis has not been used in any other thesis in this university or in any other university.

Belaynesh CHEKOL

25.01.2020

ACKNOWLEDGEMENTS

Fırst and foremost, I am deeply thankful to the Almighty, Allah, for blessing me with every opportunity He has bestowed upon me. Next, my heartfelt gratitude goes to my supervisor Dr. Numan Çelebi, for his consistent support, courage and belief throughout the entire journey to complete this research. I would also like to pass special thanks to the members of the committee, Dr. Tuğrul Taşcı and Prof. Nedim Tutkun. Dr. Tuğrul Taşcı has provided me continuous technical support, general guidance and courage which had a great role in the process. Another member of the committee, Prof. Nedim Tutkun, helped me to focus on some specifications on the research through his valuable recommendations and constructive critics during the early stages of the research.

My next gratitude goes to my husband, Sami Abdulfetah, for his love, trust and support. He has been there during the hardest times of this work reminding me that the brightest side is always at the edge of the darkest one. I thank you for trusting me wholeheartedly and for being trustworthy. Thank you for being with me happily in this journey.

My family, my father, Biadgilign Chekol, you are my pride and I would like to thank you for your prayers. My sister, Birhan Biadgilign, thank you for being my light.

Lastly, I am privileged to have a true friend, Leul Taddesse, who has been guiding me on every aspect, especially on time and resource management. Thank you for educating and helping me to acquire patience, consistency and strength.

TABLE OF CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

CC	: Connected components
ICDAR	: International Conference on Document Analysis and Recognition
GA	: Geometric accuracy
HOG	: Histogram of Oriented Gradients
SW	: Stroke Width
HSV	: Hue, saturation and value
LBP	: Linear binary pattern
MDS	: Mutual direction symmetry
MMS	: Mutual magnitude symmetry
MSER	: Maximally stable extremal regions
MVS	: Mutual Vector Symmetry
OCR	: Optical character recognition
RGB	: Red, green and blue
SIFT	: Scale invariant feature transform
SVM	: Support vector machine
SWT	: Stroke width transform
MUER	: Maximally Unstable Extremal Regions
GCF	: Global Curvature Features
CRBB	: Character recognition before binarization
CRAB	: Character recognition after recognition
WRBB	: Word recognition before binarization
WRAB	: Word recognition after binarization

LIST OF FIGURES

LIST OF TABLES

SUMMARY

Keywords: Natural scene image text, Detection, Recognition, Binarization, Scale Invariant Feature Transform, Global curvature features, Maximally Unstable Extremal Regions (MUER), Support vector machines.

This thesis deals with scene text detection and recognition as a multiple object detection and recognition problem. That is, texts that are buried within an image naturally are detected and recognized character by character. As a result, the recognition process is usually termed as segmented or cropped character recognition.

Two approaches for scene character detection are introduced. The first one is clustering based segmentation technique for multi-color scene text detection. This approach is designed to scene image texts, especially with intra-word color variance. That is, characters within the same word have distinct colors. The second approach is inspired by Maximally Stable Extremal Regions (MSER) for connected component generation. However, in this thesis, instead of stable regions, unstable regions are considered to generate candidate characters. The approach is termed as Maximally Unstable Extremal Regions (MUER) throughout the thesis.

For cropped scene character recognition, a classical approach for general object recognition is employed. That is, descriptive image features are hand-engineered and are used to train a supervised learning algorithm for recognition. Therefore, a keypoint detection and description strategy is introduced to describe the shape of character images globally. Curvature information is the primary geometric property that is employed to identify qualified keypoints. The description is dependent on major properties such as physical separation and angle between relevant image keypoints. As a classifier, SVM of various kernels is trained separately. Lastly, the description power of the global feature introduced in this thesis is compared to a well-known feature descriptor, SIFT. The results demonstrate that global shape descriptors that rely on curvature information are competitive and can ultimately lead to a better cropped character recognition.

YENI KÜRESEL EĞRİ ÖZELLİKLERİ KULLANARAK DOĞAL SAHNE GÖRÜNTÜ METNİ ALGILAMA VE TANIMA

ÖZET

Anahtar kelimeler: Doğal sahne resim metni, Algılama, Tanıma, İkilileştirme, Ölçekli Değişmeyen Özellik Dönüşümü, Global eğrilik özellikleri, Maksimum Kararsız Ekstrem Bölgeler, Destek vektör makinaları.

Bu tez, sahne metni tespiti ve tanımasini çoklu nesne tespiti ve tanıma yaklasim ile cozulmesi ile ilgilidir. Başka bir deyişle, bir görüntünün içine doğal olarak gömülü olan metinler algılanır ve karakterleri birer birer tanınır. Bu nedenle, tanıma işlemi genellikle bölümlenmiş veya kırpılmış karakter tanıma olarak adlandırılır.

Sahne karakteri tespiti için iki yaklaşım tanıtıldı. Birincisi, çok renkli sahne metin tespiti için kümeleme temelli segmentasyon tekniğidir. Bu yaklaşım, özellikle metin içi renk farkı ile görüntü metinlerini sahnelemek için tasarlanmıştır. Yani, aynı kelime içindeki karakterlerin farklı renkleri vardır. İkinci yaklaşım, bağlı bileşen üretimi için Maksimum Kararlı Ekstrem Bölgelerden esinlenmiştir. Ancak, bu tezde, istikrarlı bölgeler yerine, dengesiz bölgelerin aday karakterler ürettiği düşünülmektedir. Yaklaşım tez boyunca Maksimum Kararsız Ekstrem Bölgeler olarak adlandırılır.

Kırpılmış sahne karakteri tanıma için, genel nesne tanıma için klasik bir yaklaşım kullanılır. Başka bir deyişle, tanımlayıcı resim özellikleri el yapımıdır ve tanınma için denetimli bir öğrenme algoritması yetiştirmek için kullanılır. Bu nedenle, karakter görüntülerinin global olarak şeklini tanımlamak için bir anahtar nokta tespit ve tanımlama stratejisi tanıtılmıştır. Eğrilik bilgisi, nitelikli kilit noktaları tanımlamak için kullanılan birincil geometrik özelliktir. Tanım, ilgili görüntü kilit noktaları arasındaki fiziksel ayrılma ve açı gibi ana özelliklere bağlıdır. Bir sınıflandırıcı olarak, çeşitli çekirdeklerin Destek Vektör Makinesi'si kullanılır. Son olarak, bu tezde tanıtılan küresel özelliğin açıklama gücü, iyi bilinen bir özellik tanımlayıcı SIFT ile karşılaştırılmıştır. Sonuçlar, eğrilik bilgisine dayanan küresel şekil tanımlayıcılarının rekabetçi olduğunu ve sonuçta daha iyi bir tanıma yol açabileceğini göstermektedir.

CHAPTER 1. INTRODUCTION

In modern ages, images are considered as rich and convenient sources of relevant information. The availability of diverse imaging devices mounted on our daily used gadgets such as smartphones and tablets enable the production, processing and interpretation of images and videos easier than ever, which in turn necessitates a shift in traditional information processing paradigm. For example, an extremely large number of images are created, stored and shared through web daily. This is possible only with the help of cheap imaging devices, high capacity storage units and, of course, the advancement of World Wide Web. In addition, labels and short leading text fields in images and videos, which usually are encapsulated as captions carry significant clues to communicate without the need for explicit expression. Moreover, images can be seen as one-to-many information mappings. That is, a given image mostly has a single, discrete purpose from the source's perspective. However, the same image is likely to be interpreted differently as per the essence and the receiver. In general, information communicated through an image or a video has varying depths and hence has different level of significance for different users and destinations.

1.1. Background

Optical Character Recognition (OCR) system has evolved to transform contents of scanned document images to an equivalent digital value. The introduction of OCR system and its contribution in the simplification of retrieval, re-processing, transfer and storage of information is fundamental for most information conservation and processing tasks. In addition to scanned document images, a great deal of information is available in natural scene images, usually known as embedded text. While the task of text recognition from scanned document images is considered as a solved problem, detection and recognition of texts embedded in natural scene images is however posed as unresolved. The great composition difference between scanned document images and natural scene images is the prominent reason for the inadequacy of traditional OCR systems when applied to the problem of scene text recognition. Recently, the computer vision community has shown a rising interest towards scene text detection and recognition partly inspired by the success of object detection and recognition algorithms.

1.2. The Essence of Scene Text Detection and Recognition

Texts that are naturally buried within scene images, if detected and recognized automatically, convey profound semantic information that can be acquired to assist individual users of specific needs or as an indicator of order in multi-tasking systems. Following is an overview of potential real-time applications of scene text detectors and recognizers.

- Autonomous navigation robots: scene texts can be primary inputs for service robots to identify and follow paths that are precise to reach to a given destination within an acceptable time.

- Content-based image retrieval: A large number of images and videos are being created, transferred and stored on/through web. Efficient image and video retrieval can be obtained with the help of short descriptions on images or captions on videos that are usually included to present a compact information about a given scene.

- Instant language translation: Scene image texts are not confined to a single language. To be useful to others that are foreign to certain environment, scene text detection and recognition has a significant contribution in instant language translation and eventually improved familiarity to the scene at hand.

- Autonomous driving: In addition to service robots that are designed to be used in airports and shopping malls, there are self-driving cars that are designed to be used outdoors. Scene text detection and recognition can be very important in locating signs and guiding such cars as well.

- Industrial automation: Industries can benefit from scene text detectors and recognizers ranging from failure identification to component assembly. Even though it is at its conception, there is a huge shift in industries especially in production and maintenance that such systems are expected to have a great impact.

Because of the diversity of areas that scene text detection and recognition can involve, besides researchers from computer vision community, others, including document analysis researchers, are also participating fully in exploring the topic with the aim of designing a system which is able to perform satisfactorily as document image text recognition systems. OCR systems are able to identify text regions from scanned documents successfully [1] [2]. However, the accuracy of commercial OCR systems drop drastically where the expected input image characterized by uniform background, font, size and color is rather a scene image that naturally exhibits various complexities arising from different sources.

1.3. Challenges in Scene Text Detection and Recognition

In general, factors that are causes of major challenges experienced in scene text detection and recognition can stem from either of the following.

a. Complexities associated with the subject: - such problems are related to the scene content diversity and complexity. While variations in text color, size, orientation, font type and language are problems addressed so far, others that are caused by texture or pattern similarity between text and non-text regions require exceptional treatment and therefore deeper research.

b. Complications associated with imaging devices: - In addition to scene related issues, there are other problems introduced to an image during its acquisition. For instance, blur, which is caused by either the subject or an imaging device’s movement, is the most common. Similarly, uneven lighting, occlusion and clutter are also among properties that are typical to images collected, compiled and set as benchmark datasets.

Sample images with source scene and imaging device related problems are demonstrated in Figure 1.1. Most of these problems in scene text detection and recognition are addressed partially.

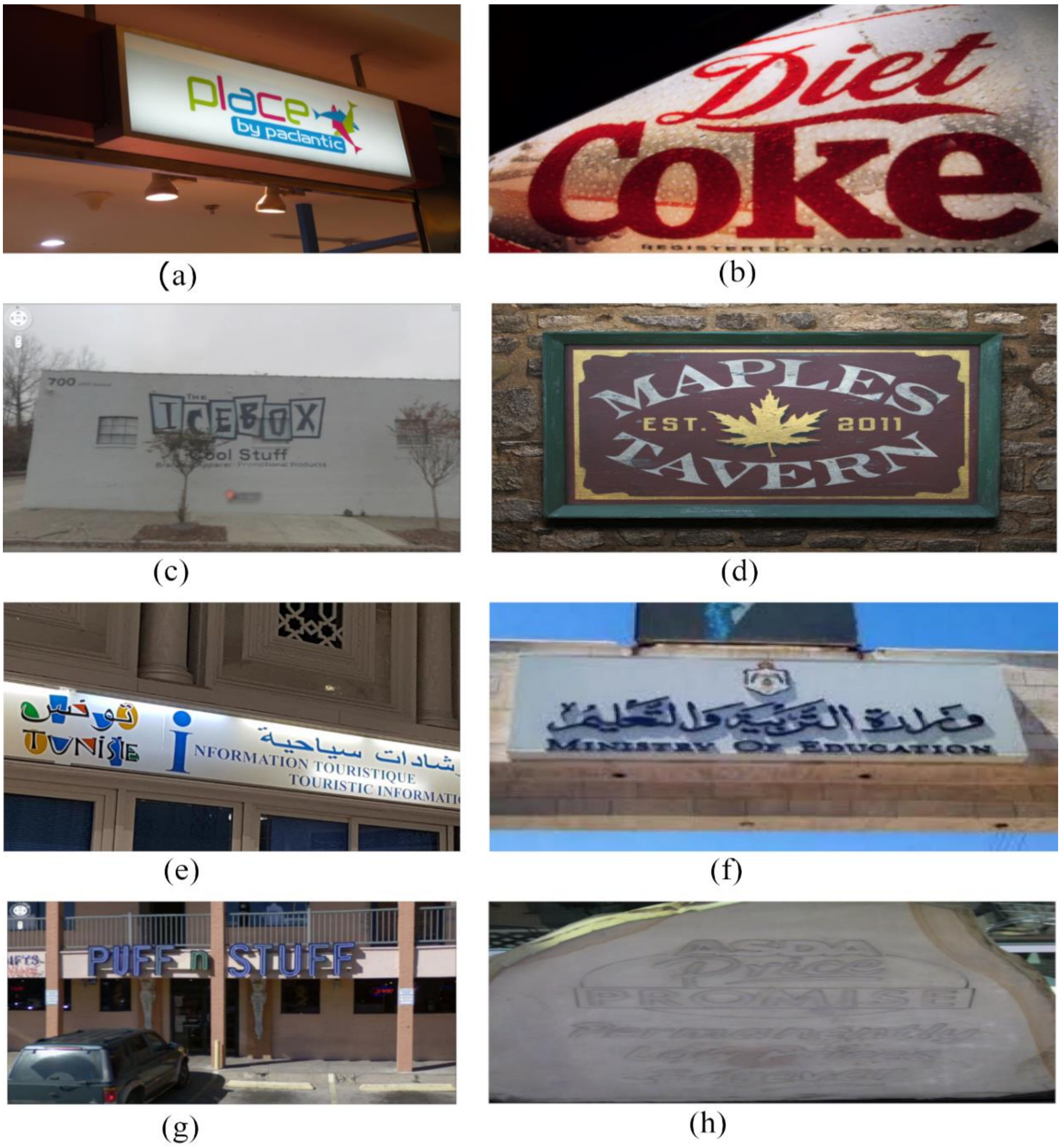

(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)

Figure 1.1. Images of scene text with :- multi-color text (a), font style(b), font size(c), multi-orientation(d), multi-language (e), blur (f), complex patterns as text(g) and low contrast(h)

1.4. Components of Scene Text Detection and Recognition

Since scene images have an unconstrained environment, controlling and avoiding disruptions is barely practical. Consequently, more intelligent and competent methods have been proposed to countermeasure each of the aforementioned difficulties. However, regardless of the tremendous effort, there is no particular, agreed-upon method that guarantees satisfactory results for general purposes. The literature on this specific field discloses an increasing number of diverse techniques. To date, the research problem is posed as unresolved and therefore it is attracting interests from various research communities, mainly computer vision and document analysis. Furthermore, regular conferences such as International Conference on Document Analysis and Recognition ICDAR2003 [3], ICDAR2005 (4), ICDAR2011 (5),ICDAR20013 (6), ICDAR2015 (7) and ICDAR2017(8) have been organized annually to follow-up advancements and report results on standard datasets. During each of these conferences, researchers are encouraged to submit a technique designed for a specific task, usually specified by the organizers. In general, the two major sub-problems specified for ICDAR conferences are scene text detection and recognition. While most researchers focus on individual tasks, others work on methods for end-to-end scene text understanding, encompassing both tasks.

Scene text detection, also referred as text spotting or localization, aims at identifying and deciding candidate character/word/text line regions. There are two main approaches that most detection related computer vision applications follow: bottom-up and top-down detection. Both of these are explained within the context of scene text as follows.

1. Bottom-up detection: - With such methods, character candidates are generated as initial outputs. Later stages include filtering non-character candidates, grouping candidates to form text lines, and recovering, to reclaim missed text pixels.

Filtering is an inevitable process in bottom-up detections. It has a significant contribution to the final detection accuracy. Therefore, a number of filtering techniques are put forward in the literature. The most well-known ones are based on learning. Supervised learners, mainly SVM and Random Ferns are trained on hand-crafted features to classify a specific image region as character, word or text line. Also, unsupervised learning algorithms including CNN are considered as suitable options for this purpose. However, most bottom-up detections rely on heuristic rules that depend on basic geometric and statistical properties. The next stage following filtering, text line formation, groups character candidates into words and words into text lines. In addition to graph-based grouping, a number of studies suggest classification and clustering as equally important options for competitive results. Putting it clearly, bottom-up scene text detection strategies are not single-shot processes. That is, there are always additional routines following detection, which if not handled properly, will definitely lead to an unacceptable result. Recovery is a procedure that is as vital as filtering. It is usually considered as the final stage in bottom-up detection approaches. Despite the presence of a range of varying proposals for recovery, context information, searching and heuristic rule-based recovery are among the most commonly used techniques. Bottom-up detection techniques are not suitable for images of low contrast and fail to endure complex backgrounds, especially if candidate characters are generated through adaptive thresholding. However, such detection strategies can be considered as best choices for scene texts where the shape of individual characters is well-preserved. Therefore, with the help of such detection techniques, texts of arbitrary shapes can be detected with much ease and flexibility [8].

2. Top-down detections: - Differing from bottom-up detection approaches, top-down detection methods generate a candidate word or a text line as a primary output. Words and text-lines are first extracted through texture, color and wavelet feature clustering. Classification using image moments and image features such as Histogram of Oriented Gradients (HOG) (9) is also an alternative to extract initial outputs. The most recent and state-of-the-art scene text detection methods employing top-down approaches are adapted from deep learning-based object detection algorithms.

With the exception of carefully designed deep-learning based scene text detection strategies, filtering false positives is imminent in top-down detection approaches as well. Therefore, similar to filtering in bottom-up detections, top-down detection techniques also rely on geometric feature based simple heuristic rules to filter out non text regions.

Although there are a number of advantages that top-down detection techniques are renowned for, the greatest one can be robust detection against complex background.

On the other hand, similar to bottom-up detection strategies, top-down detection approaches also fail to lead to good detection results in some conditions. For instance, it is clear that it is not practical to detect texts characterized by multi-orientation, font size and type irregularity. Moreover, it is computationally expensive to extract features of larger dimensions with multi-scale windows. That is, since there are no pre-defined windows to scan characters and words in scene images, it is compulsory to try a set of window sizes for each input image. That usually requires a higher computational cost. In conclusion, each of the above detection approaches are confirmed to show inconsistent performance as per the environment and there is always a tradeoff in both cases. Low precision and recall of the detection algorithm directly affects the final scene text recognition inherently.

Scene text recognition, otherwise called scene image OCR, is defined by a sequence of operations that are able to transform the results of earlier detection into word and text line equivalents of a specific language such as English, Chinese and so on. Its success is dependent on intensive pre-processing, detection and post-processing operations on candidate text regions embedded in a given natural scene image. In this thesis, a bottom-up recognition strategy is employed to label individual characters segmented from a scene image. Since a scene text has more than a single character, the problem is posed as a multiple object recognition task. Previous works on scene text recognition are completely dependent on carefully chosen text characters that are represented in a sophisticated way to train some supervised learning algorithms. However, state-of the-art methods are the results of deep-learning based algorithms.

Before the rise of deep learning to discover patterns from data automatically, determining an appropriate image feature extraction (keypoint detection) and description (keypoint description) technique requires a thorough examination and hence a set of trial and errors. Like other standard bottom-up recognition methods, the designed method starts with candidate keypoint selection that aims at retrieving distinct features from an input image. The keypoint selection is followed by a description where a single value is derived from the final key points to locally or globally distinguish each of the final key points. The choice on the description, either local or global, is also equally important as the feature detection. While there are a large number of research on the application of local feature extraction and description on object/text detection and recognition, the power of global image feature extraction and description is overlooked. Consequently, only a handful of studies are able to incorporate global image features in related tasks such as object detection and recognition.

1.5. Aims, Objectives and Contributions of the thesis

The aim of this thesis is two-folds. The first is to assess the effectiveness and efficiency of curvature based global shape features in recognizing characters segmented from scene images in comparison to local feature descriptors specifically to SIFT. The second is to assess the efficiency of connected components generated as a result of the detection of extremely unstable image elements in scene character detection. The objectives are:

- Implement SIFT specifically for segmented scene character recognition
- Implement global curvature information based feature detection and description
- Implement Maximally Unstable Extremal Region detection based connected component generation for scene text detection.
- Analyze detection and classification results obtained from the proposed features

Therefore, the thesis has two aspects. Firstly, two distinct approaches for scene character detection are introduced. Secondly, a curvature based global feature detection and description strategy is employed for SVM based scene character recognition. For scene text detection, the first approach relies on clustering where a scene image of multiple colors is transformed into binary images to locate candidate characters. The second approach, a novel connected component generation for scene character detection, is inspired by MSER. That is, initially an intensity transition based connected components are generated. The smallest units of detection are Maximally

Unstable Extremal Regions (MUER). The result of this procedure is a bi-label image, where each pixel is represented with either white; representing foreground, most regions of interest, or black; representing background, mostly other objects such as windows, grasses, fences, bricks, leaves which mostly exhibit similar texture or pattern as region of interest.

Generating connected components as MUER is followed by the determination of bounding boxes around image regions that are assumed to represent candidate characters. Since this stage usually is prone to high false positives, refinement of non-character connected components is a vital step that requires special effort. In this thesis, a Support Vector Machine (SVM) [10] classification based refinement is proposed on the histogram of curvature information computed over a given character region.

Recovery, an issue that is common to most bottom-up detections to deal with, is another serious issue that seeks equivalent exploration. During recovery, another level of processing is required to examine character regions that are rejected as a result of similarity with background regions. The result of effective recovery is improved recall that signifies how much character/word/text line is detected correctly. The comparison is with a manually annotated ground truth. That is, it answers whether the algorithm is able to locate all the characters/words/text lines included in the ground truth. Conversely, precision is dependent on how far the detection is true. Improving precision and recall at the early stages of scene text detection leads to better scene text recognition, which is the ultimate goal.

Therefore, in this study, during the early stages of recognition, emphasis is given to a global image feature extraction and description that is specifically designed for cropped scene character recognition. Similar to other common feature detection and description approaches, here also candidate keypoint selection, keypoint verification and description are integrated to get to the final feature vector.

a. Candidate key points: the transition points (MUER) obtained at the scene character detection stage are considered as candidate keypoints. Final keypoints are selected after the computation of curvature, a value that signifies how much the boundary of a character is convex or concave at each candidate keypoint. All transition points that demonstrate a higher curvature than a given threshold are selected as final key points.

b. Keypoint description: a description procedure, which is also considered as another milestone of this study, is presented to represent the regions around final key points with a vector of certain dimension. The feature description stage has a direct effect on recognition accuracy. That is, a careful feature description certainly leads to better image recognition. The description included in this study encompasses multiple geometric properties, each leading to a vector of some pre-defined size. The final feature vector is a linear combination of all individual vectors.

Lastly, the result of description, a 512-d vector, is used to train SVMs of different kernels for segmented character recognition. In addition, for the sake of performance comparison between local and global feature descriptors, SVM is trained on a very well-known local feature descriptor called Scale Invariant Feature Transform (SIFT). Recognition accuracy is computed for both types of features with a 5- fold cross-validation to determine how much SIFT and global features are effective for scene character recognition. According to the results of classification, the global image feature has a higher description power than SIFT. In addition, to be able to determine the power of the combination of local and global image features in scene character recognition, SVM is trained on the resultant feature vector obtained from linear merging (GCF+SIFT). However, it is apparent that simple integration of two different features will barely improve the classification accuracy of a learner. Devising a merging technique for two feature descriptors is beyond the scope of this research. Moreover, while the detection stage is language independent, recognition is limited to English alphabet.

The rest of the thesis is structured as follows. Chapter 2 provides a detailed revision on previous attempts to address scene text detection and recognition. Chapter 3 presents a brief description of standard datasets along with basic, distinguishing characteristics. Evaluation metrics that are used to assess effectiveness and efficiency of methods in the literature are also included in this chapter. Chapter 4 discusses preprocessing, feature detection and description, focused on the essence of these operations on classification and recognition, particularly before the introduction of deep learning techniques. Chapter 5 provides a detailed description of the designed segmentation and MUER based detection techniques. Chapter 6 consists of the scene character recognition stage which begins with feature vector generation prior to SVM training. Performance evaluation results on selected datasets are given in chapter 7. Finally, important notions drawn from the entire thesis are given as a concluding remark in chapter 8.

CHAPTER 2. LITERATURE REVIEW

Unlike a document image which mostly has a clear text laid on simple background, natural scene images embed text that is complex both in appearance and content. This property contributes to the challenges encountered in retrieving semantic information from scene images more challenging and therefore requires a deep understanding of each attribute that characterizes an embedded text. A great deal of research works focusing on individual tasks of scene text detection [(11),(12),(13),14,15], recognition [16,17,18,19] and end-to-end systems [20,21,22,23] are available. In this chapter, a comprehensive review of the most prominent scene text detection and recognition studies is presented. First, methods devised for scene text detection are explored. Since scene text recognition is the ultimate goal, equally significant attention is devoted to review some of the most influential scene text recognition studies as well.

2.1. Scene Text Detection Techniques

A comparative study of various scene text detection methods emphasizing on performances and computational complexities along with employed standard datasets is reported in Zhang et al. [24] and Zhu et al. [25]. A recent survey study, analyzing both detection and recognition techniques, is also presented by Liu et al. [26]. Both comparative and survey research allow us to reach to a conclusion that major works in object and scene text detection are entirely dependent on either manual or automated (deep learning based) feature extraction. In manual feature extraction, an algorithm identifies descriptive features of a given image or data with a substantial expert support. That is, important parameters are tuned manually. For example, edges and corners. Contrary to this, in automatic feature extraction, also called deep learning, a sub-field in machine learning and eventually AI [27] , a stack of Neural Networks are deployed to identify important patterns from a given raw data. Neural Networks have been around since the 1940s [28] . However, it is only recently that its power is demonstrated in emerging computer vision applications, especially through deep learning. However, prior to the popularity and success of deep learning in scene text detection, one detection scheme is distinguished from another mainly based on how a region of interest (RoI) is generated. The most well-known methods include connected components, sliding window and hybrid based methods. In connected components based methods, a RoI that signifies candidate characters, words and text regions is chosen based on the identification and grouping of image pixels that exhibit similar attributes such as intensity, stroke width, edge and gradient information. Since characters are primary detection units, it is reasonable to conclude that connected components based methods follow a bottom-up detection approach. Previous studies on connected component synthesis and processing are significantly dependent on major image processing tasks, mainly image segmentation. However, recent trends in connected components based detection rely on substantial information sources such as stroke width and extremal regions.

Other equally valuable studies are based on image scanning sliding windows. The literature presents us with a significant number of studies that are dependent on these methods. With sliding windows, in order to be able to capture potential text regions, a window of pre-specified size slides over the image recursively until the entire image is scanned. The intention is to locate text regions (later separated into word)] or a word at first, which, during later stages is separated into characters. A set of filtering procedures are imposed on individual regions. Unlike connected components based methods, sliding windows are able to encompass text regions of various sizes based on the scale and location of the scanning window. For example, there is a possibility of detecting the whole word with a single window. As a result, methods that rely on sliding windows follow a top-down approach. Determination of the existence of text in a given window is mostly handled with algorithms that are trained on example images that represent texts. The block diagram in Figure 2.1 demonstrates the sequence of common tasks in both bottom-up and top-down detection and recognition approaches.

Also, there are few studies that suggest hybrid methods that take up basic features from connected components as well as sliding windows. Each of these approaches (CC, SW and hybrid) has received a great attention from various communities. Major innovations and progresses are shared through some premier conferences including International Conference on Document Analysis and Recognition (ICDAR), International Conference on Computer Vision (ICCV) and Conference on Computer Vision and Pattern Recognition (CVPR).

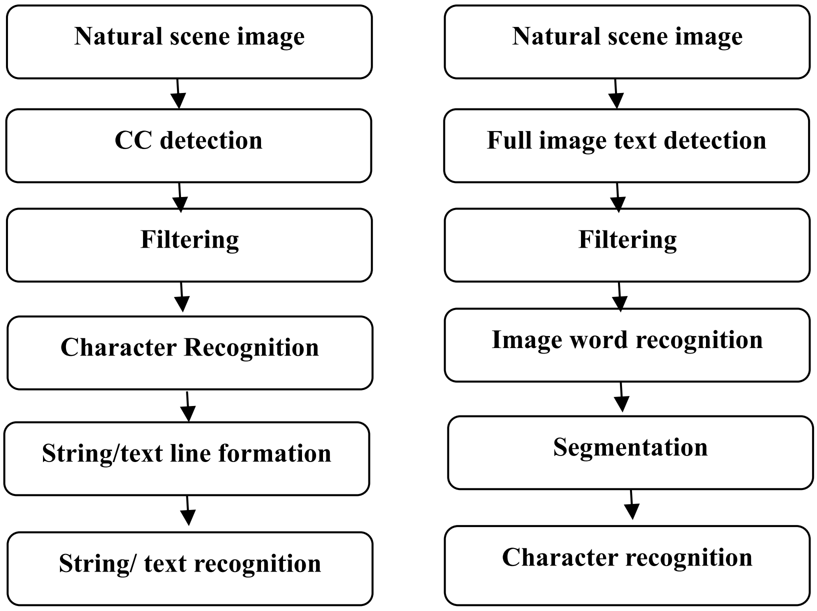


Figure 2.1. (a). Connected component and (b). Sliding window based approach

In conclusion, since most aforementioned methods are proved to be only moderately effective for general purpose [9], the literature in the field is developing and is dynamic. New insights and techniques are being introduced continuously. To reflect on the advantages and disadvantages of both methods, and to assess the various environment that each method fits, a brief review of diverse studies is included in this chapter. Lastly, a brief review of sample research studies that are considered as a breakthrough in deep learning application in end-to-end scene text recognition are also given as a sub subsection.

2.1.1. Connected component (CC) based scene text detection

Algorithms under this method follow a bottom-up approach to detect candidate text regions from natural scene images. First, a character-like image region is detected through features such as intensity, stroke width, orientation and other similar features. False positive filtering, missing character recovery and word/ text line formation are detrimental subsequent tasks. Two elementary units, Extremal Regions (ER) and stroke width (SW), are mostly used to denote connected components that represent candidate text regions.

2.1.1.1. Extremal Regions and Maximally Stable Extremal Regions based methods

An Extremal Region (ER) is a connected component (CC) whose image elements demonstrate relatively higher or lower intensities than its outer boundary image elements [1]. Maximally Stable Extremal Regions (MSER) are affine invariant, stable subset of extremal regions [29]. Both ERs and MSERs are generated from distinct image elements. Even though initially intended for object detection, the application domain is extended to many other related fields. The following section presents a review of significant research works on scene text detection that are essentially dependent on ER and MSER.

ER can be generated from various color channels. Decision on whether a given ER represents a character or not is usually supported by one or more classifiers. The work of Neumann and Matas [30] use ER obtained from the RGB,HSI and intensity gradient channels to localize and recognize scene texts. In their work, character detection is posed as an efficient sequential selection problem from the set of ERs. Probability of each ER being a character is determined based on a two stage classification. In the first stage, an AdaBoost classifier is trained on aspect ratio, compactness and number of holes. ER that has a higher character probability than a given threshold are saved for further processing. In the second stage, a Radial Basis Function (RBF) kernel SVM is trained on holes area ratio, convex hull ratio, and number of outer boundary inflexion points. The work of Neumann and Matas [31] imply MSER extraction from only two values of the HSI channel to generate text line hypotheses: Intensity and Hue. Different from most CC based works which consider a single class (candidate character), each hypothesis is classified into three classes (character, multi-character and background). Basic geometric features such as stroke area ratio, aspect ratio, compactness, convex hull area and holes area ratio are computed from each region to train a SVM that labels each region as either of the three classes. Further filtering is carried out with a local text model which relies on bottom line estimates clustering for each text line.

Similarly, Zheng et al. [11] considered an image operator that extracts extremal regions from grey, hue, saturation and Cb channels as candidate characters. Non-character regions are filtered through two SVM trained on HOG and Local Binary Patterns (LBP) to classify wide and narrow characters. Moreover, geometric features including width, height and aspect ratio are also considered to reject components that does not represent characters. The reverse operation, recovery, is based on recursive local search through similar character properties such as intensity value, stroke width, character width and height. In addition to the common tasks during detection, the work introduced a procedure that prunes repeated character structures using component trees.

On the other hand, the study of Sun et al. [14] employed component trees at the initial stages of character candidate generation. Their study benefits from enhanced ER generation methods proposed in prior related studies, thus called color-enhanced Extremal Regions (CEER). The corresponding component trees are extracted from CEER, which in turn are generated from a grayscale image, hue and saturation. NN are used both to verify text and non-text regions and text lines.

Wang et al. [1] applied MSER extracted from connected components. The study proposes a Conditional Random Field (CRF) based framework which merges results obtained from convolutional Neural Networks (CNN) that helps in determining the likelihood of an image region being a character. A context information that includes color, orientation and character shape are employed to recover text regions. False positives are filtered based on a specially designed classifier trained on binary and grey image features. Filtering and recovery are followed by text line formation and word segmentation through horizontal projection. In addition to binary and grey image features, contour and geometry based features are also considered to filter non-text regions as in Baran et al.[32]. Training a classifier to select candidate characters from the resulting MSERs is followed by merging, where selected character regions are used to generate words and phrases. The same set of features are suggested not only for filtering, but also for merging chosen MSERs.

Other similar studies suggest the combination of unsupervised learning such as clustering along with pre-determined MSERs. The work of Huang et al.[33] propose the combination of MSER with color clustering to obtain connected components. The results are filtered based on brightness contrast, connected component analysis and visual saliency map. Correspondingly, Yin et al. [34] rely on clustering algorithms to generate text candidates. As the first step, character candidates are formed from MSER algorithm. Filtering at character level is proposed with pruning algorithms. The second stage, text candidate generation, relies on morphology, orientation and projection based character grouping through single-link and hierarchical clustering. Filtering the non-text candidates is carried out through a character classification and lastly that is used to compute a posterior probability of a candidate text. Candidate text regions with high value for this parameter are removed without any further processing. The remaining text candidates are passed to AdaBoost classifier for text candidate classification.

Considering more than one color channel, Neumann et al. [23] opt to extract candidate characters as individual MSER from the R, G and B color values. Similar to most MSER based methods, character and non-character classification is done with a SVM trained on scale invariant geometric features. Text line is formed with hypothesis formation stage. The study focuses on text lines that are horizontal. Moreover, the features employed to train the classifier are rotation invariant. As a result, the learner needs to be trained on explicit examples demonstrating various orientations.

Locally adaptive thresholding is also another form of MSER extraction. The study by Gonzalez and Bergasa [16] impose CC detection from binary images obtained as a result of thresholding. The method focuses on the analysis of each MSER to locate bright and dark text regions laid on dark and bright backgrounds respectively. Basic geometric and gradient based features are calculated to determine if a given CC represents a character or not. The same features are applied to filter non-character regions. Character candidates are grouped into text lines based on a threshold value set based on the proportion of position, size, alignment and stroke width between neighboring regions. Grouped lines are accepted or rejected as a text line with a SVM classifier trained on three features: HOG, mean difference feature and standard deviation. Leveraging the object proposal technique, Gomez and Karatzats [35] proposed a three stage text extraction technique from scene images. In the first stage, a rough segmentation of the input image is obtained through MSER. Next, a text hypothesis is created with bottom-up clustering from the previous stage and finally hypotheses are ranked.

An edge enhanced MSER based detection from a contrast enhanced image is proposed by Chen et al.[36]. Prior to MSER manipulation, the paper suggests the enhancement of all MSER with the Canny edge detector based edges obtained from the grayscale equivalent of the input image. Non character components are filtered with the combination of geometric properties of MSER (aspect ratio and number of hole)] and a stroke width information calculated from a distance transform.

Yin et al. [37] proposed a MSER based scene text detection which consists of character candidate generation and pruning algorithm to filter out non character components. In addition, a character classifier which learns distance automatically filters non characters that are not eliminated through linear reduction and tree accumulation based pruning algorithm. Text candidates are constructed from character candidates with the help of single link clustering. Non text candidates are further filtered through a text classifier which computes a posterior probability of text components with the corresponding non text components. If the posterior probability is high, such text candidates are removed.

Shi et al. [38] proposed a MSER extraction that leads to two kinds of MSER; dark MSER on bright background and bright MSER on dark background. A Random Forest trained on regularity, uniform stroke width, occupation and gradient features labels each MSER as text or non-text. Unary features to analyze each MSER, color and geometric features to assess pairwise relations between MSER, is used for further false positive filtering. Later stages including text line formation are done through heuristic rules.

Koo and Kim [39] proposed a MSER based text detection that comprises of candidate generation, normalization and filtering stage. After CCs are synthesized, regions representing words are generated through CC clustering by using an adjacency relationship information obtained from an AdaBoost classifier. Unlike other CC based methods which primarily focus on CC filtering, the study suggests the elimination of such intermediate steps with the help of multi-layer perceptron trained with back propagation algorithm. Candidate word regions are first normalized with CC level information which then is followed by text/non-text classification. As a result, the method is claimed to avoid the requirement for traditional heuristic rules completely.

Ye and Doerman [40] introduced two representations called connected component appearance and consensus for MSER. These representations are prominent to decide whether individual MSER needs to be grouped into word candidates and also to determine whether the grouping results to a text or a non-text region. The appearance representation is created from HOG features and a dictionary classifier is built with a sequence of SVMs. The consensus representation is obtained from color distance, color variance and spatial distance between individual components. These two representations are integrated into a discriminative model, which is responsible to classify candidate connected components into text and non-text.

Other studies that rely on an extended version of MSER are also presented. For instance, Neuuman and Matas [41] extended the standard MSER to MSER$_{++}$ where candidate text regions are retrieved from the three channels as well as the grayscale image. Text candidates are validated with the use of region topology. Similarly, He et al.[42] present a Contrast Enhancement MSER(CE-MSE)] that extracts MSER by increasing the contrast between foreground and background. Unlike previous MSER based methods that rely on carefully designed features, the study propose the use of CNN based deep feature learning to filter non-text regions.

In conclusion, even though MSER based methods are successful and were state-of-the-art techniques for quite a longer period, after studying the most recent methods, it is evident that MSER based solutions are not ideal for many cases. For example: MSER based detection is assumed to be suitable for texts contained in images of high contrast and low illumination variance. Conversely, an intra-class intensity difference will definitely cause low detection rates. In general, scene text detection with MSER based methods is a rigorous task. This is because first, existing classification and heuristic rule based filtering are not effective. As a result, it necessitates an intelligent and exhaustive filtering mechanism as the detection mostly includes a huge number of false positives. Second, the success during earlier stages such as corner and edge detection has a direct, proportional impact on the final detection performance.

2.1.1.2. Segmentation based

Segmentation, which is also known as binarization, is a common task both in scene and document text analysis. It involves with the assignment of individual image elements to either of the two distinct regions, usually referred as foreground and background. In scene text detection, all pixels that are labelled as foreground represent the region of interest (tex)] to be detected and recognized. The accuracy at this specific preliminary stage plays a vital role at the final classification or recognition [43]. Following document image, scene and born digital image text binarization is another issue that attracts a lot of attention. Local and global thresholding based binarization techniques are the most common, especially in images with higher contrast and clear background. In both methods, the basic operation is a comparison between a given threshold and computed value within or throughout an image. Even though a wide range of scanned document text segmentation studies rely on either of these thresholding methods, there are only a handful of studies that employ local thresholding for scene text segmentation. For instance, Belhedi and Marcotegui [44], proposed a three-stage local binarization scheme suitable for whole, uncropped natural scene text embedded in a global optimization framework.

In addition to font and background complexities, text in natural scene images appear in various, random orientations. Some studies propose a binarization to address multi-oriented text detection as in Wei et al. [13]. The study propose a segmentation based method followed by a two-stage filtering procedure. In the first stage, candidate character extraction stage, a given image is transformed into grey-scale image and then a set of thresholds ranging from the minimum to the maximum grey value are applied to generate binary images. Simple rules that are formed on the basis of geometric features are proposed for the sake of coarse-level filtering. A fine-level non-character region filtering is employed with deep learning model (CNN) trained on binary images. In this method, text-line grouping is treated as a problem of pruning non adjacent graph edges from a graph representing each character.

Wu et al. [45] proposed a CC based method which rely on an adaptive multi-scale color clustering for character candidate generation and two features (Text Covariance Descriptor and HOG) to verify character and text line candidates. A SVM is trained on the entire feature set to remove non-text lines. A text covariance descriptor is a 45 dimension feature computed from features such as normalized location, mean value of the RGB and grey image, mean stroke value, occupation rate and height for the sake of filtering text lines which are classified as ambiguous during earlier classification for filtering.

Wang et al.[46] employed a superpixel segmentation based scene text detection which is composed of four fundamental units. The pre-processing step starts with the segmentation of the R, G and B values of a given scene image into superpixels. Individual connected components are generated from these superpixels by using color, local contrast and gradients of the three channels. Markov Random Field (MRF) is used with the aforementioned features. Non-text components are filtered with two stage SVM. The first stage SVM filters non text components with defined features such as occupation ratio, compactness, edge contrast, local contrast, color variance and stroke width variance. Accepted components are merged into words and text lines based on simple heuristic rule that depends on distance. The distance between two components is compared with the mean distance. Two components are determined to belong to different words if the distance between them is greater than the mean distance. In the second stage, SVM filters non text, string like components with mean probability, mean aspect ratio, mean compactness, mean occupation ratio, height variance and distance variance.

Feature based connected component generation for scene text detection is proposed in Pise and Ruikar [47]. The HOG feature is primarily used to obtain the connected components which is followed by local image binarization (Niblack's method) to extract the connected components followed by text and non-text region filtering. Two parameters, compactness and the ratio of height to width of connected components are used for this purpose.

The study by Wang et al.[48] suggest the use of a global optimization framework that embeds a local binarization scheme to generate connected components. Each text candidate with a low confidence in the confidence map is rejected. True text candidates that are rejected are recovered with the help of context information.

The study of Fabrizo et al.[49] integrate a morphological operator based segmentation for hypothesis generation and a two-step hypothesis validation for scene text detection. Initially, three SVMs are trained on Fourier descriptors, Zernike moments and polar descriptions to validate candidate characters. Later, bounding boxes of combined characters are validated with a SVM trained on HOG features.

Yi and Tian [50] introduce an image segmentation strategy for scene text detection by using uniformity of color and local gradients. Character candidates are grouped based on size differences between characters, distance between nearest characters and alignment of characters. Grouped characters are expected to form a word of at least three characters. Filtering is carried out with the help of candidate text size, aspect ratio and Euler number.

To conclude, segmentation based text detection methods are not capable of producing reliable text and background pixel assignments. Failure to do so can be attributed to a number of reasons. For instance, clustering based segmentation strategies cluster two spatially separate pixels of the same value (color, intensity) into similar groups. However, in natural scene images, texts appear in random color. Sometimes, it is possible for the text region to be in the same color as the background, provided that there is location difference. Similarly, clustering based techniques are not feasible for a real-time application that involves segmentation. This is because first, there are no rules known ahead of clustering that determines the size of the cluster. Other strategies that rely on classical thresholding are not feasible as well. There are two serious issues in thresholding. The first is finding an optimum threshold. It is very tedious, especially in local thresholding, there is no shortcut to locate an ideal local neighborhood size.

2.1.1.3. Stroke Width Transform (SW) based

The Stroke Width Transform (SW) [51] is defined as an image operator that assigns each pixel to a value that represents width of the most probable stroke to which that specific pixel belongs to. The result of this operator is a matrix with the same size as the original input, but with stroke width values as entries.

The application of SWT for the purpose of scene text detection is first proposed by Epshtein et al.[51]. In their work, they suggest the use of Canny edge detection [37] to locate edge pixels and compute stroke width for only those edge pixels instead of the whole image pixels. All boundary image elements with a corresponding opposite gradient are considered to generate a character candidate. On the other hand, word candidates are generated with the help of logical and geometric features. The same features are employed for recovering character and word candidates that are incorrectly classified as background.

Yao et al. [52] use the same concept of SWT operator on edge maps derived from Canny edge detection algorithm. The technique puts forward a four stage scene text detection method including candidate character generation with SWT operator, Character candidate analysis with a trained classifier, word candidate generation through color and geometric feature similarity among two character candidates and candidate word analysis. The analysis is based on Random Forest (RF) classifiers trained on carefully designed features such as average and turning angle, size, distance and other variations.

Likewise, a scene text detection method proposed by Risnumawan et al. [15] also relies on SWT to extract pairs of pixels by traversing in perpendicular direction within a stoke. Such pixels are tested for different types of symmetric properties such as Mutual Magnitude Symmetry (MMS), Mutual Direction Symmetry (MDS) and Mutual Vector Symmetry (MVS). This procedure is only for image pixels that are obtained from Canny and Sobel edge maps. The remaining text pixels are extracted by comparing the SIFT feature of edge pixels with the neighboring pixels. Nearest neighbors are considered as text pixels. The method filters non-text pixels with text direction.

Yao et al. [53] proposed a SWT based unified framework for both detection and recognition of texts that appear in scene images. Candidate characters are generated through grouping of image pixels using SWT and clustering. Character candidates are grouped to construct text candidates with the use of single link clustering.

Dey et al. [54] also propose a Sobel edge and Ring Radius Transform (RRT) based multi-oriented, multi-language scene text detection. The technique involves with a series of procedures to locate text regions and refine background. As per the study, convex hull is treated as the tool to avoid background pixels that are usually returned with text regions in a usual bounding box. Text components are results of merging where smaller components obtained from the edge detector are merged provided that the respective bounding boxes are intersecting.

Stroke Feature Transform (SFT), which is considered as the extension of SWT is used along with text covariance descriptors in the study of Huang et al.[55]. The scene text detection is studied as pixel, component and line level discrimination.

Lastly, it is possible to say that stroke width based techniques are fine choices for images where the edge pixels are easily and completely traceable. However, the methods are mostly dependent on the effectiveness of edge detection algorithms.

2.1.2. Sliding window based

In a sliding window based detection, the image is scanned through a window of specified size to locate a word or the entire text region within a scene image.

He et al. [56] proposed a Cascaded Convolutional Text Network (CCTN) that departs from a traditional CNN based sliding window approach. In this case, a heat map is generated for individual characters and a two stage detection with a customized convolutional network is stated to be efficient to handle multi-shape and multi-scale text regions. The first stage is a coarse text network which directly outputs a per-pixel heat map informing the probability and location of text. In the second stage, fine text network, outputs two heat maps for each cropped regions representing central lines and text line separating areas.

Region proposal methods that were initially designed for object detection are also employed in scene text detection. Jaderberg et al. [57] use a text based image retrieval system based on an object-agnostic region proposal (word proposal) using edge boxes and a sliding window detector to evaluate bounding boxes and train AdaBoost classifiers on aggregate channel features . The paper suggests a random forest classifier to filter false positive words.

Wang et al. [21] uses multi-layer neural networks with unsupervised feature learning to detect and recognize scene texts. A 32x32 size window slides over the entire color image to determine if there is a character candidate in each image patch. Post processing operation including beam search and NMS are also considered as an integral part of this particular study.

A combination of strokes and sliding window based searching is proposed in Neumann and Matas [58]. Initially, strokes are generated as a result of a convolution operation on an image intensity gradient with a set of oriented bar filters. Candidate regions are generated from strokes and bounding boxes are returned. Unlike traditional sliding window methods which search candidate text regions throughout, following stroke generation, the method considers only those bounding boxes that consist of at least one stroke. Text line formation problem is handled with partial ordering where regions, preceding and following regions need to be in certain order.

Even though sliding window based methods are optimal choices for text detection from low contrast scene images, there are critical issues that get in the way of further consideration. Firstly, sliding window based methods are not appropriate choices for multi-oriented scene text detection. Also, there is no any agreed upon procedure that can be used to determine an ideal initial window size. As a result, these methods are computationally expensive and it is impossible to define and set distinct, robust rules to discern non-text region.

2.1.3. Hybrid approach

Merging distinguishing characteristics of CC and SW based approaches, a few studies suggest a third approach otherwise called hybrid approach, to natural scene image text detection.

Pan et al. [12] explained the complementary property of CC based methods to sliding window based methods. Similar to previous CC and sliding window based methods, the study puts forward a three stage technique where initially the image is scanned with a 16- by-16 window to determine the text confidence. Image patches of high probability from the text confidence map are converted into binary images based on Niblack's local binarization algorithm [59]. Conditional Random Field (CRF) is applied on the resulting binary image to preserve character components. Clustering of individual character components based on minimum spanning tree algorithm is employed to generate words. Text lines are also formed by grouping word components with an energy minimization model. Similarly, Zhao et al. [60] coupled CC and SW methods loosely. In the first stage text candidates are determined with a SW which is then followed by a CC based character components and final text candidate extraction. That is, in the first stage, text confidence map is created from Partial Differential Equations (PDE). From the resulting confidence map, text region candidates are extracted based on local binarization. The next stage, character candidate extraction is carried out with the help of color clustering and simple rules on the candidate text regions. Non-text regions are filtered through AdaBoost classifiers.

A more robust method which combines the power of CNN with MSER and sliding window methods is proposed in Huang et al.[61]. Similar to other MSER methods, the first task is extraction of MSER from a given image. Next, a trained CNN assigns a score to each MSER. Text lines are finally formed from all MSER whose confidence scores are considered to be acceptable. In the second stage, sliding window is used with CNN to determine two components, however are returned as one component. As stated briefly in the original work, a given component is further processed for decomposition if it has a positive score and a high aspect ratio.

A sliding-window based detection through Viola-Jones style cascade of Haar wavelets is combined with CC extraction from a segmented image in Bissaco et al. [62] . However, there is no explicit elaboration on filtering and recovery techniques utilized.

Other studies such as Zhang et al.[63] propose a scene text detection technique that completely diverts from both connected component generation and sliding window based detection. That is, unlike previous detection units (characters and words), it is able to locate text lines. The paper suggests the use of symmetry features obtained from gradient direction and magnitude computed on a stroke. Also, a CNN classifier is included to filter non-text candidates and a NMS to reject redundant detections. Likewise, Coates et al. [64] present a multi-stage scene text detection that includes unsupervised feature learning, convolution based feature evaluation, spatial pooling based feature dimension reduction and linear classifier based detection.

In conclusion, CC, SW and hybrid methods are successful methods if preceded by practical pre-processing and competent post-processing routines. Moreover, with these methods, the detection and recognition are dealt as completely independent tasks. However, state-of-the-art detection techniques are basically dependent on deep-learning.

2.1.4. Deep learning based scene text detection approaches

Departing from the traditional character and whole text detection methods, recent trends on scene text detection focus on the application of deep learning algorithms to overcome limitations that emanate from hand crafted features and complex post-processing tasks. There are a number of efficient deep-learning algorithms proposed in object detection and recognition. Some researchers opt for these algorithms to handle scene text detection problems by considering words or text lines as objects. Reported accuracies are promising and far better than CC and SW based state-of-the-art approaches. Some of these works are examined briefly as follows.

In scene text detection, in addition to the most prevalent challenges, the appearance of text in an arbitrary orientation is also getting attention. Accordingly, Ma et al. [65] suggests an arbitrary orientation text detection algorithm that encompasses a Rotation Region Proposal Network (RRP) to generate a random proposal for text instances and bounding box regression, Rotation Region of Interest (RROI) to project arbitrary text proposals into feature maps and lastly a two layer network to classify regions as text and background.

Qin and Manduchi [66] proposed a cascaded convolutional neural networks word-level text spotting approach. The first convolutional neural network (TextSegNet) is a FCN which uses both local and global context information to detect text of arbitrary shapes and size. The second network (WordDetNet) analyses the resulting text segment obtained from the previous network. It is a two-stage detection procedure. Similarly, Gomez et al. [35] proposed a three-step object proposal based text extraction method.

Qin et al. [67] approached the scene text detection problem with multibox processing and semantic segmentation. Multibox processing results to a number of bounding boxes representing text candidates. In addition, classification probability of each pixel is obtained from a softmax layer added at the output layer of the semantic segmentation. In addition to these stages, the paper proposes a module called bounding box enhancement module that merges the results obtained from the multibox processing and semantic segmentation.

He et al.[68] proposed a deep direct regression detection mechanism for multi-oriented scene texts. The paper suggests a fully convolutional neural network that performs feature extraction, multi-level feature fusion and multi-tasking to generate a pixel-level classification and vertex coordinate representing a text bounding box. In addition, a one step-post processing called recall non maximum suppression is proposed to reduce the number of densely overlapped quadrilaterals for a word or text line.

Jaderberg et al.[69] build on other CNN based works by generating a whole text saliency map using text and non-text classifier on the entire image, thus avoiding down-sampling at a pixel-level or segmented character proposals.

Departing from the traditional rectangular bounding box used with CNN, Liu and Jin [70] use a new CNN named Deep Matching Prior Network (DMPNet). The method consists of three stages to detect multi-oriented scene texts. These include a quadrilateral window based text detection, regression to predict text with compact quadrangle and auxiliary loss function based text position regression.

Busta et al.[71] adapt a real-time object detection algorithm: You Look Only Once (YOLOv2) [72]. Unlike other methods that rely on the same architecture, the study avoids one of the limitations of YOLO architecture, rotation sensitivity, by generating a number of rotated bounding boxes at each last convolutional layer than a single bounding box.

Another study by Gupta eta al.,[73] also present a Fully Convolutional Regression Network (FCRN) based scene text detection by computing text specific features at various layers, which finally are supplied to the regression networks. The study suggests that the speed of CNN based detections can be improved by decreasing the frequency of CNN evaluation for an image.

Zhong et al.[74] use a region proposal and detection based unified framework with Fully Convolutional Neural Network (FCN). The region proposal process uses Inception Region Proposal Network (Inception-RPN) to come up with word candidates bearing text characteristics such as aspect ratio of width and height. The text detection network involves with the classification of candidate words into text and non-text with the help of multi-level region of interest pooling. An iterative bounding box voting scheme is included to improve the recall and precision of the proposed technique.

Similarly, Jiang et al.[75] apply a RPN to generate axis aligned bounding boxes that are aimed at detecting text regions of various orientations. Features obtained from multi-scale ROI pooling are used to train a modified Fast-RCN which is able to classify text regions, predict inclined boxes and lastly refine false detections. Final detection results are obtained after Non Maximal Suppression (NMS) is employed on inclined bounding boxes.

Others such as Zhou et al. [76] use FCN to predict a word or text line as rotated rectangles and quadrangles, which are filtered with NMS during later stages. Unlike previous works which rely on heavy post-processing operations, this paper introduces only one post-processing task which is thresholding.

A ResNet-50 based shared convolution networks for both scene text detection and recognition is proposed in Liu et al. [77]. These networks produce low and high level semantic feature maps which in the end is used by one convolution to generate a dense pixel-level text prediction. Thersholding and NMS are carried out as post-processing to produce final detections. Text-like patterns are filtered through online hard example mining and the experimental results reported in the paper show an improved F-measure.

Similarly, Dai et al.[78] proposed a multi-oriented scene text detection method that combines an accurate region proposal with an instance-aware segmentation technique. The researchers argue that a deep CNN model which has feature, ROI and pixel-level text instance generation is better than other region based methods such as Faster-RCNN and SSD. Also, NMS and minimal quadrilateral generation are involved during later stages of processing. Likewise, Yang et al. [79] use a special type of ROI , Position Sensitive-ROI (PSROI) that can be deformed to be able to handle scene texts of arbitrary orientations as it encodes spatial information. The combination of PSROI and an adaptable convolution layer is a distinguishing feature. Moreover, the study benefits from GoogleNet’s inception model to handle texts of various scales, aspect ratio and orientation.

Inspired by SSD, other studies such as Liao et al. [80] present Textboxes, a FCN based scene text detector that predicts bounding boxes for word candidates with only one neural network. The accuracy of the detector is improved further with the help of a text recognition module and a standard NMS.

The idea of region proposal is reformed to meet basic requirements of text and is referred as text proposal in Bazazian et al.[81]. The paper suggests a detection technique that has two basic aspects: the text proposal algorithm and a Fully Convolutional Network (FCN). The text proposal algorithm is responsible for generating bounding boxes surrounding candidate text regions. This stage is carried out through image segmentation based on MSER. The FCN on the other hand generates heat maps which in the next stage of filtering is used to re-rank locations obtained from the text proposal algorithm. Zhang et al. [82] too, combined FCN with CC methods, where the power of MSER is exploited to find character components which in turn is considered as an input to determine the orientation for text lines. Two independent FCN are used to generate a pixel-level salient map for text regions and character centroid prediction for the sake of false positive filtering.

The work in Deng et al.[83] avoids location regression by suggesting instance segmentation than semantic segmentation which is employed by most previous deep-learning based methods. The CNN is trained to predict pixel-wise text/non text labels along with the corresponding link between a specific pixel and its eight neighbors. CC are generated as a result of connection among neighboring image pixels which are determined to be in the same instance. Unlike other methods which rely on bounding box regression, in this particular work, bounding boxes are directly obtained from semantic segmentations. Geometric properties of the final bounding boxes are used to filter false positive detections.

He et al.[84] introduce a text detector that generates a bounding box for individual words regardless of scale and orientation with a single shot. The paper suggests the complete avoidance of a layer of multiple FCN and post processing operations with the exception of common NMS.

Wu and Natarajan [85] propose a three class scene text detection based on FCN. The classes include the usual text and non-text classes along with a new class, border, introduced in the study. One of the fundamental focus is to avoid multiple post-processing operations and also diverge from object detection algorithms. The base of divergence is level of homogeneity, where text regions are assumed to contain a relatively high homogeneity than object classes.

In addition to FCN and Region Proposal Networks, Convolutional Recurrent Neural Networks (CRNN) are also employed for scene text detection. Li et al.[86] suggest the reduction of intermediate processing steps experienced in other studies including candidate character generation, merging and word separation by designing a single forward pass RNN. The RNN is responsible to encode feature maps of varying lengths resulted from ROI pooling. The same features, of the same size, are used for both detection and recognition, one complementing the other.

In conclusion, deep-learning based scene text detection strategies are state-of-the art methods in terms of detection accuracy and detection speed. However, as a data-thirsty model, the efficiency if highly dependent the training and testing dataset diversity. This issue is currently addressed with synthetic images. Unfortunately, it is clear that it is only partially that Ican replicate events in the wild with the aid of synthetic images. Moreover, such models require a high performing end, which eventually incurs limitation in application scope.

2.2. Scene Text Recognition

Recently, similar to scene text detection, scene text recognition is also an important research topic. While there is a significant number of research and techniques, there still is a need for a more robust strategy for recognition which is completely independent of language model and lexicon support. Extensive research on the literature shows that scene text recognition methods can generally follow either of the following recognition units: characters, words and sequence.

2.2.1. Character classification based scene text recognition

Methods under this category focus on the classification of individual characters or matching each character to a template. Later, individual recognitions are combined to form words. These type of methods require a great care while designing features and choosing classifiers. In addition, higher classification accuracy require the use of certain language models. For character based text recognition, classifiers need to be trained on large size annotated character datasets, which is inadequate currently for real-world applications. To overcome scarcity in an annotated character datasets for scene text recognition, De Campos et al. [87] present a dataset of English and Kannada characters extracted from natural scene and synthetic images containing frequently used fonts. Using the dataset, various image features are tested to assess if object recognition frameworks are able to countermeasure common problems that commercial OCR systems encounter when reading text from scene images. Besides the evaluation of existing features, a feature that is an aggregate of visual words retrieved from each class is also tested. Nearest Neighbor (NN), SVM and Multiple Kernel Learning (MKL) based classification is put forward for individual character recognition. Other studies such as Sinha et al. [88] suggest pre-processing operations before training an ensemble machine learning for character classifiers. The work employs image pixels and HOG as features to train NN, Random Forest and Extra Tree classifiers to recognize individual character images. Similarly, features that are variants of HOG such as CoHOG and CovCoHOG are also employed for segmented character recognition as in Su et al.[89]. Adapting HOG, which is most widely used in object recognition tasks, the study suggests the extraction of co-occurrence of HOG between neighboring pixels to derive a more informative and robust feature from various image patches. Final features that are designed to train linear SVM are fetched with the help of average pooling.

Conversely, other studies apply a bi-classification directly on candidate characters obtained from previous stages. Neumann and Matas [23] devise a scene text recognition which is based on character/non-character classification of individual ERs. From each MSER, a feature of 200 dimension is extracted and fed to a RBF kernel

SVM for classification. In order to differentiate similar upper and lower case letters, a feed-back loop is included where the height of a character recognized at that instant is compared with the height of a template predefined for the same character. In another study, Neumann and Matas [31] propose a refinement based recognition at character level. Connected components whose aspect ratio is determined to be less than a given threshold are assigned a Unicode label and a learner is trained on these values. All the resulting labels form a cyclic graph and the final word or text is obtained from the optimal path of the graph. Likewise, Yao et al.[53] presented a character classification that uses histogram of component level features along with Random Forest classifier for scene text recognition. In order to correct errors in classification, a dictionary of most frequently searched words from a search engine is introduced. Case disambiguation is carried out through mean score and relative size computation between the initial and the rest of the letters.

A K-nearest neighbors (KNN) based character classification based scene text recognition is proposed by Gonzalez and Bergasa [16]. The method assumes a binary image of a character from which edge pixels are detected. For each edge pixel, direction of gradient is computed and lastly represented with histogram. The final feature obtained, a 128-d vector, is fed to KNN for classification.

A linear classifier trained on Fisher Vectors (FV) for scene character recognition is proposed by Shi et al.[90]. Inspired by the success of these features in other classification tasks such as image classification, the study proposes a FV generated from Gaussian Mixture Models (GMM) along with spatial information that encodes the position of local features. Similarly, Wang et al.[91] apply FV to reach to a global representation of a character image by encoding character part co-occurrence features. Initially, they extracted convolutional activations from CNN to describe character parts locally. Later, Multi-order Co-occurrence Activation (MCA) fetches the multi-order co-occurrence information between character parts. The final feature vector is used to train CNN and ultimately recognize characters within scene images.

While most character classification based recognition assumes a segmented character, others such as Elagouni et al.[19] propose a direct character recognition from a given input image such that the image is scanned with windows of various sizes to determine if a particular image patch represents a character or not. Two convolutional networks defined as window recognizer CNN and character recognizer CNN are designed to classify each patch as character/non-character and assign labels to image patches that are classified as characters respectively. In order to avoid confusions as a result of window border overlapping and filter recognition, they included graph and language model. Also, Wang et al.[21] propose an end-to-end character based recognition where image patches representing characters are located and recognized simultaneously with the help of CNN. The classifiers are trained on features retrieved with an unsupervised feature learning algorithm. The final word is formed as a beam search based combination of individual character recognitions.

Hidden Markov Models (HMM) trained on the combination of both local and global image features and a 128-d Local Gradient Histogram feature are also used for character recognition in Roy et al.[92]. The features are extracted from a sliding window whose path is estimated prior to feature extraction.

Another well-known texture feature Local Binary Pattern(LBP) was also employed for scene character recognition in the study presented by Yang and Yang [93]. The study implies improved recognition as a result of an improvement of the existing LBP with the addition of useful edge and pixel information. The label ‘Improved LBP’ is coined for the version introduced in the paper.

A study by Higga and Hotta [94] introduce Tangent Distance (TD) to achieve invariance towards transformations such as rotation, shift and scaling. The resulting vectors are combined with Local Subspace Classifier (LSC) for scene character recognition.

Liu and Lu [95] propose a Markov Random Field (MRF) based scene character recognition that relies on local interest points and spatial information that represents global structures. Both sources of information are considered for template character images and input images. Lastly, a number of one-versus-one classifiers vote for matches and the label for the character image is determined.

In addition to common classifiers, the work of Bissaco et al.[62] demonstrate the application of deep neural networks for scene character recognition. The networks are trained on pixel values and HOG features. Other geometrical features and language models are also employed to improve the recognition accuracy.

2.2.2. Word classification based scene text recognition

In order to avoid typical problems encountered in character classification based scene text recognition, essentially, grouping recognized characters into words, a number of studies propose a method which classifies an entire image representing a word. Like the character based recognition, in this case also, features from word images are extracted and fed to classifiers to find the final word label. Mostly, a lexicon of words and metadata are provided to the model to enhance the recognition.

Su and Lu [17] proposed a Recurrent neural Network (RNN) based scene text recognition. The input for the proposed technique is assumed to be a word image. Sequential image features are extracted from the given input based on HOG. These feature vectors are fed to a multi-layer RNN to be classified into the corresponding words. The connectionist temporal classification (CTC) technique is employed to filter out the most accurate label based on the match between the recognition result from RNN and a list of words in a lexicon.

2.2.3. Sequence based scene text recognition

Deterring from the aforementioned methods, more holistic approach for scene text recognition is the trend recently. Mainly, with the adoption of speech recognition techniques, contextual information in scene texts is captured as a sequence recognition.

Inspired by the efficiency of 1D Connectionist Temporal Classification (CTC) in speech recognition, Wan et al. [96] adapted vanilla CTC model into 2D CTC and employed it for scene text recognition. A network generates a probability distribution map through softmax output layer and a path transition map.

Yang et al.[18] propose an ensemble of Deep Neural Networks (DN)] that has three fundamental stages including base classifier generation, classifier combination and pruning. In base classifier generation, Convolutional Recurrent Neural Networks [CRNN] are used to generate neural network components as text recognizers. The sequence probability of components are predicted with CTC which is set as an output layer. The results of ensemble are filtered with Genetic Algorithm (GA) based pruning.

Gao et al.[97] also rely on CTC to produce an output after character selection, concatenation and repetition, and blank space removal. The study shows how semantic information can be utilized at explicit (word level) and inexplicit (character level) to recognize every text within a cropped image with the need for pre-defined lexicon. The paper introduces a supervision enhancement branch that is used to improve recognition by letting individual levels support one another.

The study of Wang et al. [98] points out the significance of spatial information preservation while generating feature vectors from 2d feature maps. Unlike previous sequence based recognition techniques, the paper benefits from the spatiotemporal relations between image elements by incorporating an attention mechanism into ConvLSTM and character center masks.

Likewise, the work of Cheng et al.[99] paid attention to the essence of attention networks in sequence based scene text recognition. The study incorporates focusing networks to avoid the usual problem of “attention drift”, where feature vector areas and image regions are not aligned correctly.

Other studies including Zhan and LU [22] propose a pre-recognition stage of rectification aiming at enhancement of an image exhibiting perspective and curvature distortion. The study aims at better recognition accuracy through text pose estimation and text distortion correction, where state-of-the-art deep learning based recognition algorithms demonstrate inefficiency.

Deep Recurrent model that uses LSTM to recognize word images as a set of labelled sequences is proposed by He et al.[100]. An input word image is transformed into an ordered sequence with the help of CNN. These sequences are recognized by Deep Recurrent Model which relies on LSTM to store specific order information. The researchers claim the feature to be robust against major image distortions. In addition, an explicit lexicon is not required.

CHAPTER 3. STANDARD DATASETS AND EVALUATION METRICS

The rank of a given scene text detection and recognition technique is determined based on the comparison results with state-of-the-art techniques. To be able to compare methods reasonably and reach to a decision, it is necessary to rely on common, similar standard datasets and performance assessment techniques. The following section provides brief introduction on the most widely used standard datasets in scene text detection and recognition along with performance assessment metrics and protocols.

3.1. Benchmark Datasets

A number of standard datasets are made available to the public through different communities to assist researchers follow up advancements in state-of –the-art scene text detection and recognition techniques and set a base-line. Moreover, such datasets are resources on which the efficiency and effectiveness of a new technique is measured and compared to other profound techniques. Even though most of the datasets have common characteristics, some of them are intended to address specific issues such as multi-orientation and multi-language. While the majority of these datasets are real-world scene images, the rest are a collection of synthesized images. That is, images are created synthetically to reduce efforts required in manual acquisition and supplement real scene images. Table 1 summarizes the most widely used scene text detection and recognition datasets with basic attributes describing each.

Table 1.1. Benchmark datasets in scene text detection and recognition.

Datasets	Training	Testing	Languages	Orientations	Annotations
ICDAR 2003	258	251	English	Horizontal	word and character
ICDAR 2011	229	255	English	Horizontal	word only
ICDAR 2013	229	233	English	Horizontal	Word and character
ICDAR 2015	1500		English	Multi oriented	word
ICDAR 2017	7200	9000	Mulit-language	Multi oriented	word
MSRA-TD500	300	200	English and Chinese	Multi oriented	text line
COCO-TEXT	43,686	20000	English	Horizontal	word
SVT	350		English	Horizontal	word
IIIT 5K	2000	3000	English	Horizontal	word and character
SynthText	858,750		English	Horizontal	word character and text line
Synth90 k	9 Million		English	Horizontal	word
Chars 74k	74,000		English and Kannada	Multi oriented	character
OSTD	89		English	Non-horizontal	text-line
NEOCR	659		Multi-language	Multi oriented	
KAIST	3000		Multi-language		
SVHN	600,000		English	Horizontal	Character
Total Text	1555		Multi-language	Curved text	
ICDAR-2019 ArT	5603	4563	English and Chinese	Arbitrary Shaped Text	
Cute80	80			Curved	

- ICDAR2003[1]: It was released for the ICDAR2003 Robust Reading competitions which aims at robust text locating, character and word recognition. The challenge was to locate text regions, recognize characters and words from focused scene images, images that are taken primarily to capture a text field.

- ICDAR2011[2]: The ICDAR2011 Robust Reading competition released a dataset of born digital images, (web and Email) characterized by low resolution in addition to scene images provided in earlier competitions (ICDAR2003).

- ICDAR2013[3] [6]: The ICDAR2003,2011 datasets are claimed to contain duplicates. Furthermore, the given ground truths were not precise as well. For ICDAR2013 robust reading competition, these issues are handled and as a result the ICDAR2013 dataset has less duplicates with improved ground truths.

- ICDAR2015[4] [7]: While all the aforementioned datasets comprise of images of scenes that are acquired with focus on textual fields, for ICDAR2015 roust reading competition, a dataset of images with incidental text is introduced. That is, all the images in this dataset are acquired with no special intention onto text regions.

- ICDAR2017[5]: For ICDAR of 2017, the robust reading competition is extended to encompass methods that are able to read texts of more than one language in a single scene image. Since previous datasets are collection of images with texts of a specific language (English), it was compulsory to include a dataset that shows images containing texts in more than one language.

[1] http://algoval.essex.ac.uk/icdar/Datasets.html

[2] https://rrc.cvc.uab.es/?ch=1&com=downloads

3 https://rrc.cvc.uab.es/?ch=2&com=downloads

[4] https://rrc.cvc.uab.es/?ch=4&com=downloads

5 https://rrc.cvc.uab.es/?ch=8&com=downloads

- Total-Text[6]: A dataset comprising images of diverse nature. It includes both web and scene images with texts of multiple languages.

- ICDAR-2019 ArT[7]: It has 10,166 images imported from other three datasets (Total-Text, SCUT-CTW1500 and Baidu Curved Scene Text). The dataset is called arbitrary shaped text dataset since there are images of horizontal, multi-oriented and curved texts.

- Large-scale street view text[8]: This dataset has images captured from complex real time scenarios on streets.

- MSRA-TD500[9]: A dataset of web and scene images created to track progresses in multi-oriented text detection from indoor and outdoor images.

- COCO-TEXT[10]: This dataset is based on MS COCO dataset. Images are taken with no special attention directed towards text regions. As a result, it is mainly employed to measure performances of methods that are designed particularly for incidental scene text detection.

- SVT[11] [101]: This dataset is a collection of highly variable and low resolution images from Google street view.

- IIIT 5K[12] [102]: images in this dataset are retrieved with Google search using key query words including house number, billboard and signboards.

[6] https://github.com/cs-chan/Total-Text-Dataset/tree/master/Dataset

[7] https://rrc.cvc.uab.es/?ch=14&com=downloads

[8] http://bjyz-ai.epc.baidu.com/broad/download?dataset=lsvt

[9] http://tc11.cvc.uab.es/datasets/MSRA-TD500_1

[10] https://bgshih.github.io/cocotext/

[11] http://tc11.cvc.uab.es/datasets/SVT_1

[12] http://cvit.iiit.ac.in/projects/SceneTextUnderstanding/IIIT5K.html

- SynthText[13] [73]: It is a dataset that contains words and scene images which are not together naturally. About 8 million synthetic words are artificially rendered on 800,000 scene images.

- Synth 90k[14] [103]: It is a dataset of synthetic images consisting of all 90 thousand words found in the English language dictionary.

- Chars 74k[15] : IT is a dataset of characters including handwritten, synthesized and scene characters.

- OSTD: It is a dataset of 89 scene images that embed non-horizontal text regions extracted from various environments.

- USTB-SV1K[16]: A dataset that contains 1,000 multi-view and multi-orientation images directly extracted from Google Street View.

- CurvetText (Cute80)[17]: It consists of 80 natural scene images with curved text lines.

3.2. Performance Evaluation Standards and Metrics

Besides reliance on common benchmark datasets, the performance of a given detection and recognition method is also evaluated with metrics that are accepted as standard measures. It is essential to reference performance improvements and innovations according to these fundamental requirements. Since the essence of scene text detection

[13] http://www.robots.ox.ac.uk/~vgg/data/scenetext/SynthText.zip

[14] https://www.robots.ox.ac.uk/~vgg/data/text/mjsynth.tar.gz

[15] http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/

[16] http://prir.ustb.edu.cn/TexStar/MOMV-text-detection/.

[17] http://cs-chan.com/downloads_CUTE80_dataset.html

is completely different from that of scene text recognition, the measurement and assessment is also significantly different for these phases.

3.2.1. Scene text detection performance evaluation standards and metrics

In classical object detection, there are two types of matching operations that are widely used as a standard to determine the stance of new detection techniques with regard to state-of-the-art methods. These are: object matching based on the object area intersection and object matching based on Euclidean distance between detected and ground truth boxes with the corresponding centroids [104]. Most scene text detection evaluation standards are derived from object detection standards that are based on matching. The possible outcomes as a result of matching estimates (detection result) and targets (ground truths set mostly by human annotators) are explained as follows. Interpretation also varies from one metric to another.

The first one is zero-to-one matching. In this case, the detection algorithm returns a bounding box around an image region whose corresponding ground truth is not defined or given. In scene text detection, such results are returned in situations where the detection algorithm returns a bounding box that surrounds a region which has similar textural appearance with text regions. For example, grasses, bricks, fences and so on. These errors are causes for low precision. On the other hand, one-to-zero matching is the exact opposite. In such cases, a detection algorithm fails to return a bounding box around a region of interest (object, text) that has a corresponding ground truth entry. In scene text detection, this occurs when a detection algorithm fails to detect a character or string that appears on a scene image. It causes an algorithm to exhibit low recall. Among other matching types, a successful detection can be expressed with one-to-one matching. In one-to-one matching, an estimate from an algorithm (detected bounding box) and a target (ground truth) are equivalent. Also, one-to-many matchings can be considered as successful detection for bottom-up detection approaches where the detection algorithm produces more than one estimates for a region defined by a single ground truth. This occurs when characters forming a specific string are detected individually. It is termed as splitting error. On the contrary, top-down detection approaches usually lead to a single output representing multiple characters. In such cases, the detection algorithm returns a single bounding box that encloses multiple targets. This type of matching is termed as many-to-one matching. A text line or text region in general, which has more than one string is bounded by a single bounding box and consequently is seen as a merging error. Lastly, many-to- many matching, similar to one-to-one matching, can signify successful detections.

The above-mentioned matching types are common and some of them are even attention catching issues. Therefore, there are various metrics that are introduced to compromise and count the contribution of detections that belong to one of the three (one-to-many, many-to-one, many-to-many) matching types and eventually improve detection performance. In general, an algorithm evaluation metric is worth of consideration if there are mechanisms to cope with and count detections that encounter merging and splitting errors.

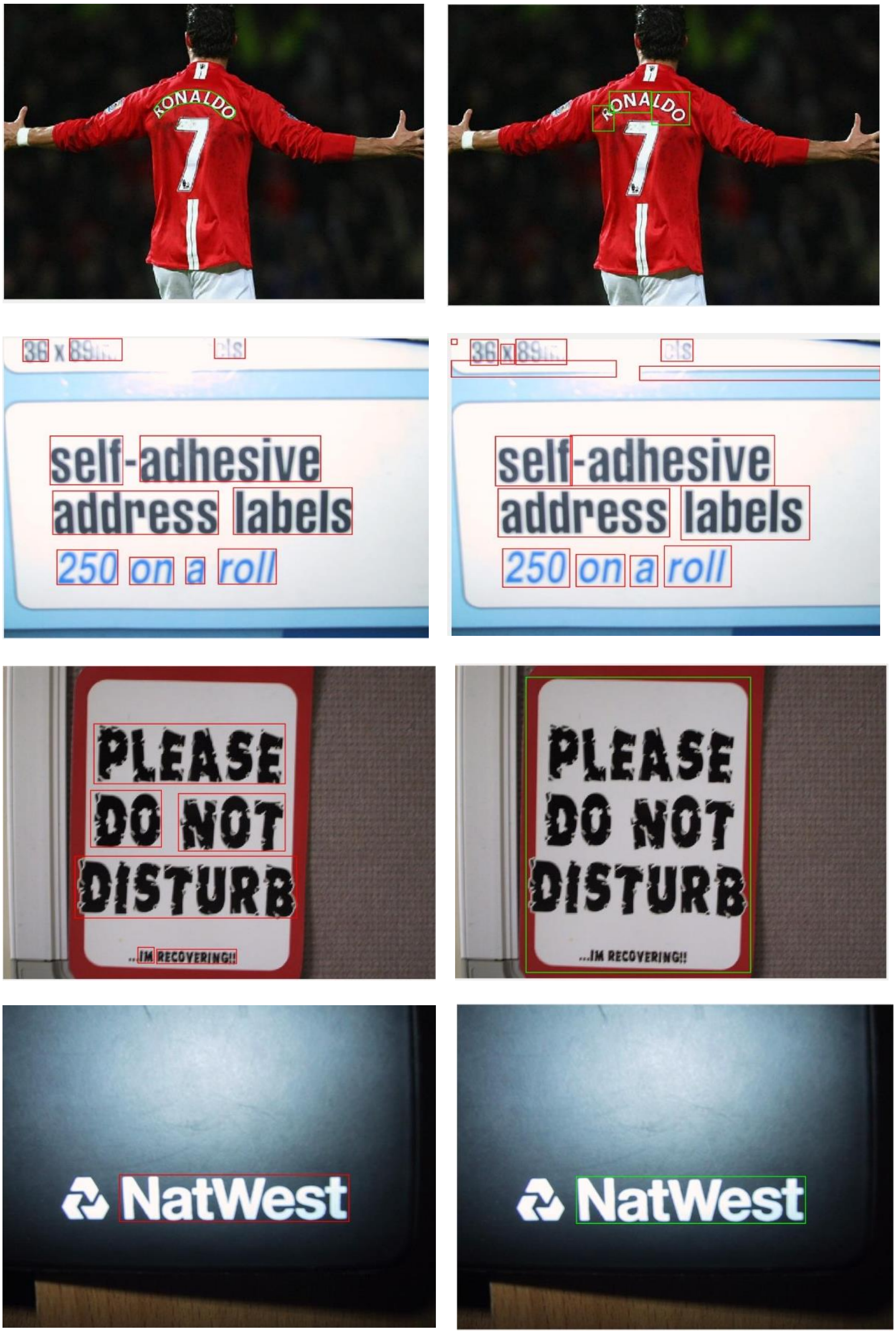


Figure 3.1. Matching types.

Various performance assessment protocols handle the matching problem and derive corresponding evaluation metrics such as false positive rate, false alarm rate, detection rate accuracy, precision, recall and F-measure. However, precision, recall and F-measure are the most popular evaluation metrics in scene text detection. Consequently the computation of these values varies from one protocol to another. Based on the aforementioned matching strategies, the following section gives a brief overview over the standards based on these metrics.

a. ICDAR 2003 and 2005 standard: This standard is derived from object detection performance evaluation PASCAL EVAL's Intersection over Union (IoU) with little modification. The IoU is calculated with equation 3.1.

$$IoU = \frac{Area(E \cap Gt)}{Area(E \cup Gt)} \tag{3.1}$$

Where *E* is detected bounding box and *Gt* is the given ground truth. In ICDAR 2003 and 2005, this value is calculated with equation 3.2.

$$IoU = \frac{Area(E \cap Gt)}{Area(\min(RECTANGLES(EandGt)))} \tag{3.2}$$

The ICDAR 2003/2005 standard punishes one-to-many, many-to-many and many-to-one detections assigning a zero match score, and consequently underestimating the performance of a particular algorithm. For each ground truth, the match with the largest IoU value is selected. Hence, the best match for a ground truth (Gt) in a set of detections (Ds) is expressed as equation 3.3.

$$M(Gt, Ds) = \max(\{Gt, d)\} | d \in Ds) \tag{3.3}$$

Where *Gt* is ground truth, *Ds* is set of rectangles detected with a given algorithm and *d* is a specific detection considered at a time.

Metrics, precision, recall and F-measure are calculated with equation 3.4, 3.5 and 3.6 respectively.

$$precision = \frac{\sum rd \in Ds \text{ bestmatch(rd:Gt)}}{|Ds|} \tag{3.4}$$

$$bestmatch(rd:Gt) = \max_{i=1....|Gt|} \frac{2 * area(rd \cap Gi)}{area(rd) + area(Gi)}$$

$$\mathrm{Re}call = \frac{\sum rg \in Gt \text{ bestmatch(rg:Ds)}}{|Gt|} \tag{3.5}$$

$$bestmatch(rd:Gt) = \max_{i=1....|Ds|} \frac{2 * area(rg \cap Di)}{area(Di) + area(rg)}$$

$$f\_measure = \frac{1}{\frac{\alpha}{precision} + \frac{1-\alpha}{recall}} \tag{3.6}$$

Where α is set to 0.5 to control the relative weights of *precision* and *recall*, *rg* and *rd* are instances from ground truth set and detected rectangle sets respectively.

b. ICDAR 2011 and 2013: Since the previous protocol disregards performance by rejecting many-to-one and one-to-many matchings, the organizers of the annual ICDAR robust reading competition devised a mechanism to improve the previous evaluation protocol by considering such detections and compute a match score. This protocol is adopted from the evaluation protocol of Wolf and Jolion [105] that presents advantages such as information on how many text rectangles are false and correct detections, a relatively easy interpretation of a detection quality, support to splits and merges and easy scale up to multiple images without losing its power and ease of interpretation. In this protocol, the precision and recall are calculated according to equation 3.7 and 3.8 respectively.

$$precision = \sum_i \left(\frac{MatchG(Gi, D, tr, tp}{|G|} \right) \quad (3.6)$$

$$recall = \sum_j \left(\frac{MatchD(Dj, G, tr, tp}{|D|} \right) \quad (3.7)$$

Where the *matchD* and *matchG* are the functions that consider different types of matching and |*G*| and |*D*| are number of ground truth and detected rectangles respectively, *tr* is area recall which is calculated as the ratio of area of intersection to the area of ground truth rectangle and *tp* is area precision which is calculated as the ration of area of intersection to area of predicted rectangle.

Area precision and recall in for ICDAR 2011/2013 is set to 0.4 and 0.8 respectively. Both matchD and matchG have values in [0, 1]. One-to zero and zero-to one detections return zero for both functions, one- to-one detections return one and one-to-many detections (split errors) and many- to- one (merge errors) return values in between.

c. Evaluation Protocol of MSRA-TD500. A more considerate performance assessment protocol, particularly for scene text detection algorithms designed for multi-oriented scene text is proposed by Yao et al [52]. The protocol adopts the concept of minimum area rectangles [106], which are claimed to be much tighter than axis-aligned rectangles. However, computing the overlap ratio between the detected rectangle 'D' and the ground truth rectangle 'G' using minimum area rectangle is tedious. Therefore, in [60] 'D' and 'G' are rotated around their centers and the overlap ratio is computed using axis-aligned rectangles with equation 3.9.

$$overlapratio(G, D) = \frac{area(axis\_aligned(G) \cap axis\_aligned(D)}{area(axis\_aligned(G) \cup axis\_aligned(D)} \quad (3.8)$$

With this form of evaluation, if the angle between the estimated and ground truth rectangles is less than π/8, and the overlap ratio is greater 0.5, the estimated rectangle is counted as a correct detection. Multiple detections of the same text line are taken as false positives. Similar to PASCAL visual objects challenge evaluation protocol in [107], precision is calculated as the ratio of true positives to the total number of detections. On the other hand, recall is calculated as the ratio of true positives to the total number of ground truth rectangles. To avoid some limitations incurred by IoU methods, some researchers build up on it to customize it towards goal oriented detection. For Example, Liu et al [108] proposed a tightness aware Intersect-over-Union (TIoU). Recently, Lee et al. [109] argues that current state-of-the- art protocols fail in addressing issues such as granularity, multiline and character incompleteness. In order to fill this gap, the paper proposes a protocol that is based on instance-level matching and character-level scoring.

3.2.2. Scene text recognition performance evaluation standards and metrics

Unlike detection which is evaluated against the coordinates of a rectangle that surrounds the detected text, recognition is measured at two discrete levels. Even though recognition metrics are less punishing than detection, there is a requirement for special consideration when the recognition is measured based on word recognition accuracy. The ratio of the total number of correctly recognized words to the total number of words in the ground truth, called as word recognition accuracy, is a commonly used scene text recognition evaluation metric [110]. Fine-level metrics such as character count to determine how many characters are recognized accurately, is also another metric to evaluate character recognition based methods.

In ICDAR 2015, a standard edit distance metric based evaluation protocol is used to evaluate word recognition of incidental text. In this metric, each addition, deletion or substitution is counted equally. For example, if a scene text has one word composed of five characters as ‘crowd’ and the recognition result is ‘crown’, the edit distance will be four.

CHAPTER 4. IMAGE PREPROCESSING, IMAGE FEATURES AND CLASSIFICATION :PRE-DEEP LEARNING

Before the adoption of the power of deep learning techniques into text detection and recognition, the feasibility and accuracy of detection and recognition methods were highly dependent on image features that are tuned manually with the help of feature extraction algorithms, hand crafted features. Transformed into some descriptive form, such features are used with various learning algorithms (supervised and unsupervised) at coarse and fine levels. Moreover, during the period when the concept of connected components and stroke width transforms were state-of-the-art scene text detection methods, image features were the primary tools to determine and filter out regions that are not in the range of interest. The process which involves with the determination of image pixel/s to be a distinctive property of an image and hence representing it with a unique description is termed as feature extraction and description respectively.

4.1. Image Pre-processing

The primary task in many pattern recognition and classification tasks is image pre-processing. The ultimatum is to prepare the image for intermediate level image analysis tasks such as feature extraction and description for higher level domain specific applications including classification/recognition and image retrieval. In scene text detection and recognition, most researchers employ various types of filters on original images to either suppress noise and irrelevant content or enhance important features such as edges and corners. Based on the size of image region considered for transformation (a pixel, small neighborhood, whole image), image pre-processing methods can be classified into four: Pixel brightness transformation, geometric transformations, neighborhood based transformations and image restorations [111].

1. Pixel brightness transformation: The intensity of each pixel in the image is transformed into a new value obtained from the original value itself (For example, when the intensity is inverted, added, subtracted, mapped to some calculated value). The domain is limited to each individual pixel and as a result these operations do not affect and modify the image spatially.

2. Geometric transformation: such transformations are considered as essential to remove distortions that were introduced during image acquisition as pixels in the original image are set to a different spatial location with the help of a transform function. These include rotation, translation, scaling and skewing.

3. Neighborhood based transformations: Unlike pixel brightness transformation, which is entirely dependent on a particular pixel's intensity, such transformations are based on the intensities of neighboring pixels too. Such transformations are critical when noise suppression is required. Operations such as averaging in the literature are suggested to be effective in reducing impulse noises and small stripes. Based on the function that a neighborhood is processed, these transformations can be divided into linear and non-linear filters. Linear filters modify pixel values directly through the use of a sliding window. For instance, mean filter. On the other hand, non-linear filters do not possess a uniform weight to transform individual pixels. For example, Median filter.

4. Image restoration: It is the process of filtering image distortions caused by camera motion, subject motion and other noise to transform an image into its original form through a priori and posteriori information about the nature of the degradation.

4.2. Feature Extraction and Description

Image features are metrics that are used to represent a given image at a higher level [112].Feature extraction and description is the process of identifying representative characteristics from a given image or image region and representing it with either a single compact representation (global feature) or a set of vectors that signify more information sources such as multiple regions of interest (local feature). It is a fundamental step in most text detection and recognition techniques both at the preliminary stages of detection and final stages of recognition. It is a representation transition from a binary, gray or color image to a quantitative data, which is usually a *'nx1'* vector, *'n'* suggesting the length of a particular feature. Therefore, the objective is to transform an input image into a form that is more compact and convenient to train algorithms. Classification to recognize characters, words and other objects, and clustering are dependent on the final representation. Different applications in pattern recognition require different forms of feature types and description techniques. In general, a given image can be described with either local or global features.

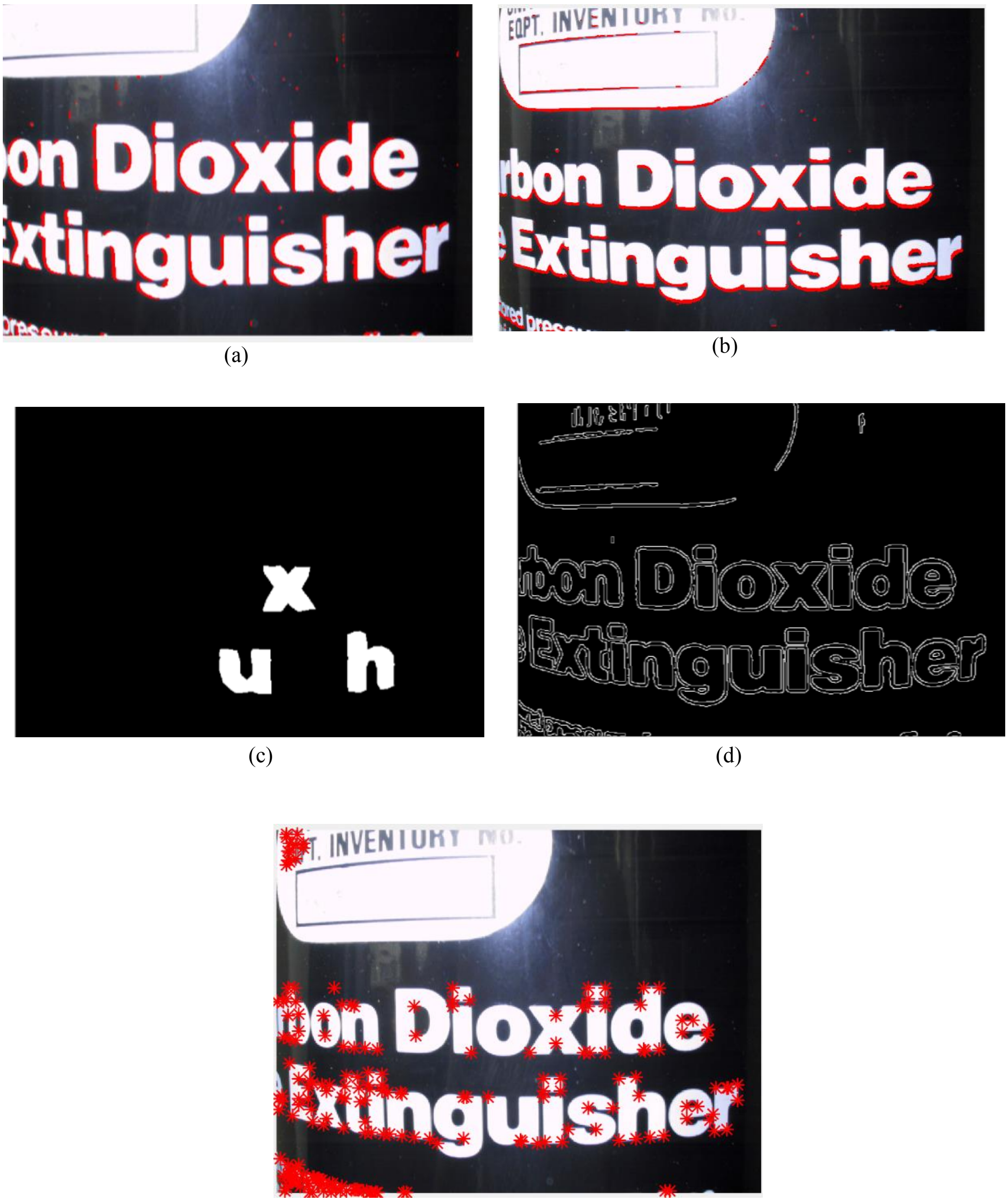
Local features are image patterns that have different value from a nearest neighborhood [113]. These features are expressed with changes in intensity, color and texture. Most local features are extracted from varying size, independent image sub-regions. Consequently, each region description results to a feature vector of different length. Before employing these features for classification and clustering tasks, a method is required to normalize the feature dimensions to make uneven feature lengths invariant throughout an image. These features are robust as far as clutter and occlusion. The most important local features include, but is not limited to, edges, corners and regions [114].

a. Edges: are a set of curved line segments which are obtained as a result of image points that show discontinuities; intensity transitions from high to low or vice versa.

b. Corners: are points at which the image has a significant change in intensity along all directions. A corner can also be expressed as an intersection between edges.

c. Regions*:* are closed set of connected points possessing similar intensity values surrounded by multiple regions.

Images under figure 4.1 illustrate color transitions, interest regions, edges (boundaries) and corners that can be transformed into some form of description for a required computer vision task.



(a) (b)

(c) (d)

(e)

Figure 4.1. High and low transitions (a) and (b), interest regions (c), edges and boundaries (d) and corners (e).

Global features are derived from the whole image. As a result, a single vector is a result of global image description. Global image feature descriptors are easy to compute and lead to a compact representation of the whole image. Such features are usually used to represent image color, shape, homogeneity and texture.

a. Image color: an attribute that is usually the easiest image property for human eyes to extract relevant content from a given image. color moments, color histograms, color coherence vector and color correlogram are among the most commonly used color descriptors [115].

b. Image shape: A shape can be described with different parameters including center of gravity, axis of least inertia, digital bending energy, eccentricity, circularity ratio, elliptic variance, rectangularity, convexity, solidity, Euler number, profiles, hole area ratio[116].

c. Texture: provides a measure of various properties such as smoothness, coarseness, and regularity [112]. It is also one of the most vital features that are used to identify regions of interest in an image [117].

4.2.1. Feature detectors

'Feature detector' and 'feature extractor' are the two words which are used equivalently to refer to techniques or algorithms that are used to retrieve image features such as points, edges, regions and corners. Generally, local image feature detectors are based on, but not confined to, contour curvatures, intensity, color, models or machine learning.

Contour curvature based detectors such as Harris detectors are mainly designed to detect corners which appear as junctions between two edges or contours and eventually suggest an intensity change across all directions. Other detectors such as intensity based detectors are based on convolution of an image intensity with kernels of various sizes and forms to discover important information about an image. For instance, edges and fast transitions. Difference of Gaussian (DOG) and Laplacian of Gaussian (LOG) are popular intensity based feature detectors. Others detectors such as model based detectors are not dependent on gradient or other properties, rather are dependent on brightness comparison. For example BRISK: Binary Robust Invariant Scalable Keypoints [118] , FREAK : Fast Retina Keypoints [119] and ORB: Oriented FAST and rotated BRIEF [120]. Model based features are commonly used in matching based computer vision tasks. Segmentation is also considered as a powerful tool to detect regions which in later stages are processed and filtered to be used as distinctive image features. For example, MSER based scene text detection methods were the most successful detection strategies up until the biggest shift into deep learning. Machine learning algorithms such as neural networks, decision trees and genetic algorithms are also exploited to detect corners and interest points. Other methods which are used to detect important features such as edges rely on differentiation. For instance Sobel and Canny edge detectors.

4.2.1.1. Essential properties of feature detectors

1. Fast to compute: The ultimate goal of feature extraction and description is image matching for various applications such as recognition, retrieval and image registration. Since most of these operations involve with a large image collection, the feature extraction and description needs to be feasible to be used in real-time applications.

2. Descriptive: The features are expected to be representative without losing generality and requirement for domain specific knowledge.

3. Memory efficient: features are required to be compact enough and be stored in a relatively small storage area both during at computation and at the end when referred.

4. Robust: the features used to represent an image or region of interest needs to be invariant to rotation, scaling, skewing and translation. Moreover, it is

required to be robust towards noise that might be introduced during image acquisition and transfer.

5. Repeatable: the requirement that the same features are expected under different viewing conditions.

6. Unique: Image feature representing the same interest point/region on different images is expected to be the same.

7. Accurate: Image features should be localized accurately. In image matching applications, accurate feature detection leads to familiar results on the query and test images.

8. Generality: The ability of a feature detector to detect features that can be used in different applications. The detector should not be confined to a specific application scope.

9. Quantity: The feature detector should be able to locate most, if not all, of relevant features. The larger the size of the feature number, the better the image is represented.

4.3. Feature Description and Classification

The feature extraction stage is totally concerned with the discovery of key points and regions of interest within an image that convey relevant information about transitions, significant changes and homogeneity. While this stage is a fundamental step towards efficient domain specific application, to be employed in for a given purpose, it should be coupled with an effective description that is resilient to local and global image deformations. Feature description is the process of encoding the location, orientation and scale of the detected features (key points, regions, corners) through the extraction of diverse properties describing a set of neighboring image elements. Descriptors such as SURF, SIFT and HOG are among the most widely used ones. SURF [121] is a scale and rotation invariant descriptor whose detection is based on a Hessian matrix measurement. HOG is another popular feature descriptor that quantizes the occurrences of gradient orientation within image regions specified for a specific application. Even though it is aimed at object recognition, recently it has attracted attention from various researchers in different fields including document analysis. Since the thesis involves with the application of SIFT for cropped character recognition, its concept and underlying principles are studied in depth.

4.3.1. Scale Invariant Feature Transform (SIFT)

SIFT, an algorithm designed by David G. LoI[122] to detect and describe key points of a given image is a scale, rotation and translation invariant feature descriptor that demonstrate a great performance in image recognition and matching. It is mainly employed in applications where objects or parts of an object are required to be uniquely identified and matched. However, initially SIFT was intended to locate a number of similar interest points from two images irrespective of image scale differences so that they can be used for image matching, registration, mosaicking or object recognition. SIFT is praised for the following prominent properties.

a. Locality: features are local, so robust to occlusion and clutter (no prior segmentation).
b. Distinctiveness: individual features can be matched correctly to a large database of features obtained from many objects.
c. Quantity: a large number of features can be generated for even small objects.
d. Efficiency: when applied for areas which SIFT is claimed to be ideal, it gives a close to real-time performance.
e. Extensibility: can easily be extended to wide range of differing feature types, with each adding robustness.
f. Robust: invariant to scaling, translation and rotation. In addition, it is partially invariant to illumination changes and affine or 3D projection.

4.3.1.1. Phases in SIFT feature extraction and description

The original version of SIFT encompasses four major subtasks to identify image keypoints and reach to a single representation for each image element that is determined to be a keypoint. The following subsections explain each subtask with the help of sample character images and results obtained from its application, particularly in a scene character recognition context.

1. Detection of local (Scale Space) Extrema: The first step in finding a local extrema is to create a number of blurred versions [five in the original paper] of the input image by convolving it with a 2D Gaussian function with varying standard deviations (σ), initially set to 1.6. Successive images are obtained by blurring the image with a multiple of this value and a parameter 'k' whose value is given as$\sqrt{2}$ .Scale space is the usual term that describes the result of the algorithm at this specific stage (Refer Figure 4.2).

The scale space detection is followed by calculation of Difference of Gaussian (DoG) between two consecutive images in the scale space. When employed on two or more dimensions, the DoG operator is assumed to approximate the Laplacian of Gaussian (LoG) with far better computational cost [123].

Assume $D(x, y, \sigma)$ as the difference between two Gaussian blurred images (immediate neighbors), but with a different σ. The DoG value is computed as follows with equation 4.1.

$$D(x, y, \sigma) = G(x, y, \sigma) * I(x, y) - G(x, y, k\sigma) * I(x, y) \qquad (4.1)$$

Where $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{\frac{[-(x^2+y^2])}{2\sigma^2}}$ and $G(x, y, k\sigma) = \frac{1}{2\pi k\sigma^2} e^{\frac{[-(x^2+y^2])}{2k\sigma^2}}$ represents the Gaussian function. On a different notation, it is given as the result of the difference between two Gaussian functions convolved with the image. The results, however, are equivalent.

2. Extrema selection (local minimum or maximum): Considering a specific pixel *p(x, y)* each time, the intensity of this image element is compared with neighboring pixels *(p(x,y+1),p(x+1,y),p(x-1,y),p(x,y-1),p(x+1,y+1),p(x-1,y-1),p(x+1,y-1)* and p(x-1,y+1)) in the same octave and 9 pixels in images of octaves before and after the current octave. That is, the comparison is done for all eight neighboring pixels and the corresponding pixel in a given octave too. (Refer Figure 4.3, a pixel surrounded with red is the one to be compared, all the pixels given in white boundaries shall be considered for comparison).The image pixels with the minimum or maximum value than the remaining 26 pixels are stored as keypoints or interest points in that scale and space. It is called extrema as it indicates either a drastically low or high intensity value. All image points that are chosen during this stage are inputs for the second stage which is filtering, or usually called localization.

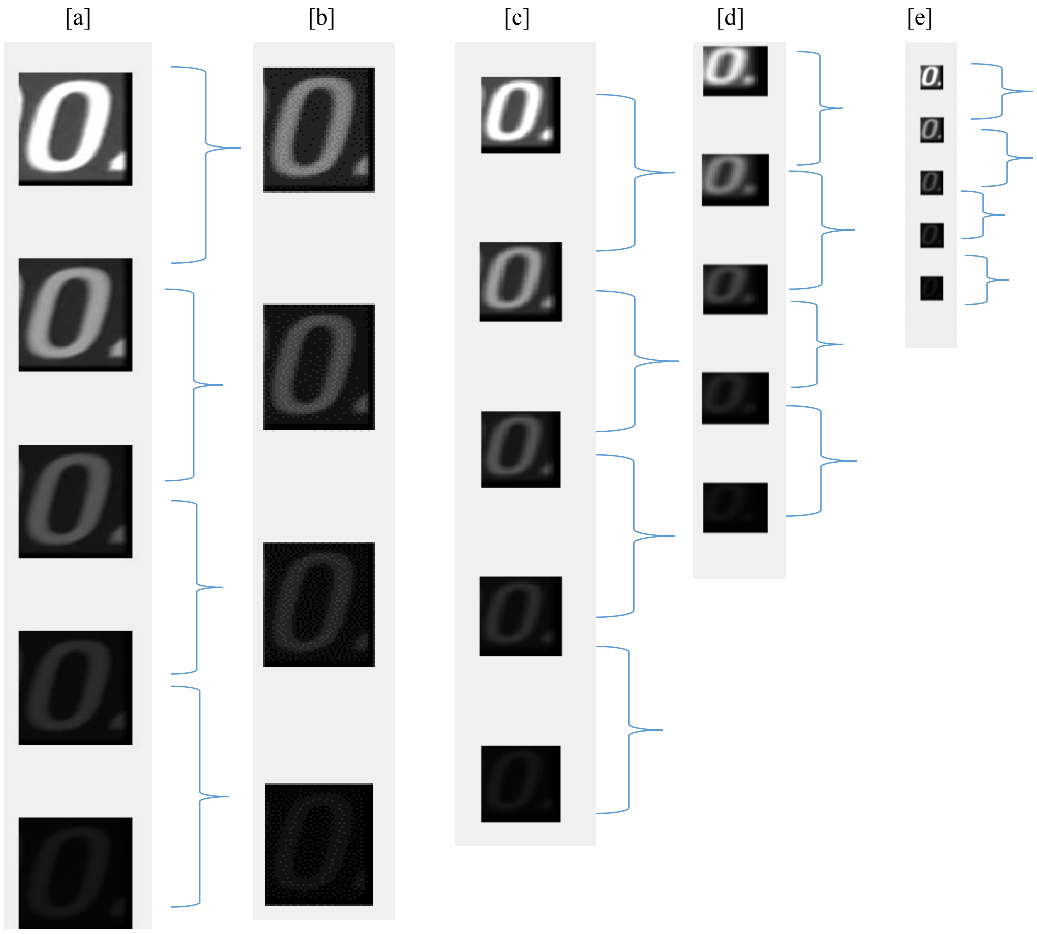


Figure 4.1. Demonstration of octaves, scale space and DoG, first octave (a), DoG image from first octave (b), second octave(c), third octave(d) and fourth octave (e)

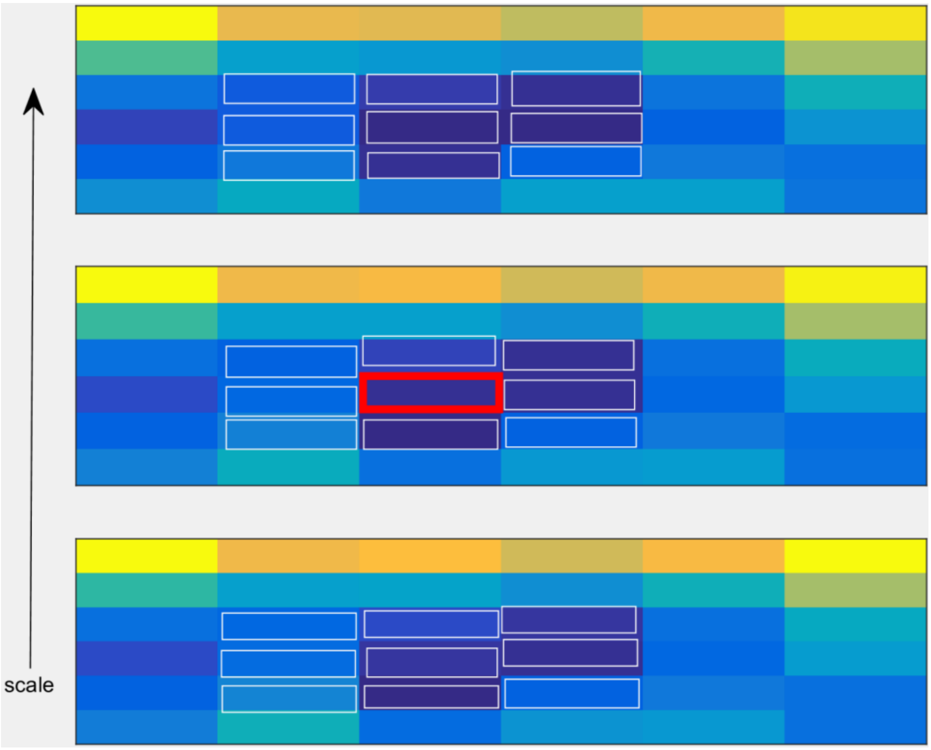


Figure 4. 2 . A pixel (red boundary), 8 neighbors in the same octave and 18 pixels from one above and below octaves.

3. keypoint localization: Among the minimum and maximum points detected in the previous stage, it is highly possible for some of the pixels to be unstable and not localized properly. In order to be able to represent only the most stable pixels, it is important to examine if a chosen pixel passes some contrast threshold or determine if a chosen pixel lies on an edge or not. For this specific purpose, two tests are designed and applied for each image element in the extrema: low contrast and edge test. An image pixel in an extrema set is said to be low contrast if the magnitude of intensity at that specific pixel in the DoG is less than a threshold value. Such pixels are removed in the first localization process. Next, image elements in the extrema set which lie on edges are detected and removed based on the value of the principal curvature computed at the location and scale of a given pixel. If the ratio of the principal curvature is greater than a threshold, it is an indication of the presence of some form of instability. As a result, such pixels are removed from the extrema set.

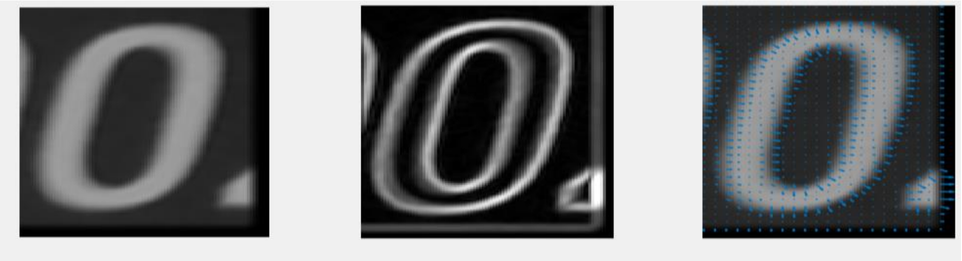
4. Orientation assignment: The next stage is to determine the scale at which all the stable points are detected, which is fundamental to make features scale invariant. Rotation invariance is achieved through assigning each stable image element an orientation. To do this, gradient direction and magnitude of a Gaussian blurred image is computed using the following equation 4.2 and 4.3 respectively.

$$m(x,y)=\sqrt{L(x+1,y)-L(x-1,y)^2+L(x,y+1)-L(x,y-1)^2} \tag{4.2}$$

$$\theta(x,y)=\tan^{-1}\frac{L(x,y+1)-L(x,y-1)}{L(x+1,y)-L(x-1,y)} \tag{4.3}$$

Where $m(x,y)$ is for gradient magnitude at a given location, $\theta(x,y)$ is gradient direction and L is a Gaussian blurred image on which the keypoint is detected.

Sample gradient magnitude and direction computed on Gaussian blurred image (in the third octave) is illustrated with Figure 4.4.



(a) (b) (c)

Figure 4.3. Gaussian blurred image (a), Gradient magnitude (b) and Gradient orientation(c)

To assign orientation for keypoints, orientations around the keypoint are sampled and the dominant orientation (s) is assigned as an orientation for a particular keypoint and this operation is repeated for all keypoints that are determined to be stable and properly localized (Refer Figure 4.5).

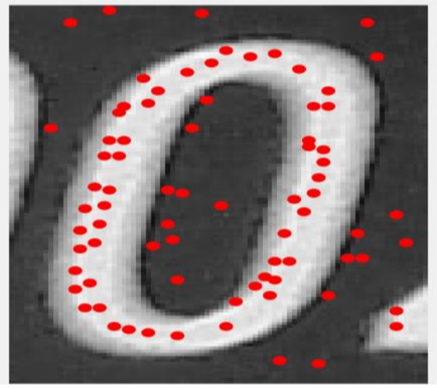


Figure 4.4. Keypoints (in red)

The number of neighboring pixels to be examined in order to determine the dominant orientation and ultimately the keypoint's orientation is dependent on the scale. The bigger the scale, the more the number of neighboring pixels considered at a time. The orientation assignment stage is fundamentally a representation of a given keypoint's orientation with the help of a histogram of orientations of all considered neighboring pixels. The direction of every pixel is in the range between 0 and 360 degrees. A histogram is constructed with 10 degrees gap resulting in a 36 bin histogram. The gradient is the core factor to determine the amount to be added to the bin. The higher the gradient magnitude, the higher the amount added. Once this step is completed for all detected keypoints, the dominant orientation is the peak of the histogram. The corresponding bin number to which a particular peak resides is assigned as a direction for the keypoint.

5. Keypoint description: In order to be able to use detected features from earlier procedures in a real time application that involves with classification and clustering tasks, feature detection should be followed by an interesting, inevitable transformation, known as feature description. Since it is equally important as feature extraction, usually, a significant amount of effort is devoted during this stage as well. To achieve a unique description for each keypoint, a 16x16 size image around the keypoints is considered for each keypoint, which during subsequent operations is further divided into 4x4 size windows to produce a total of sixteen regions.

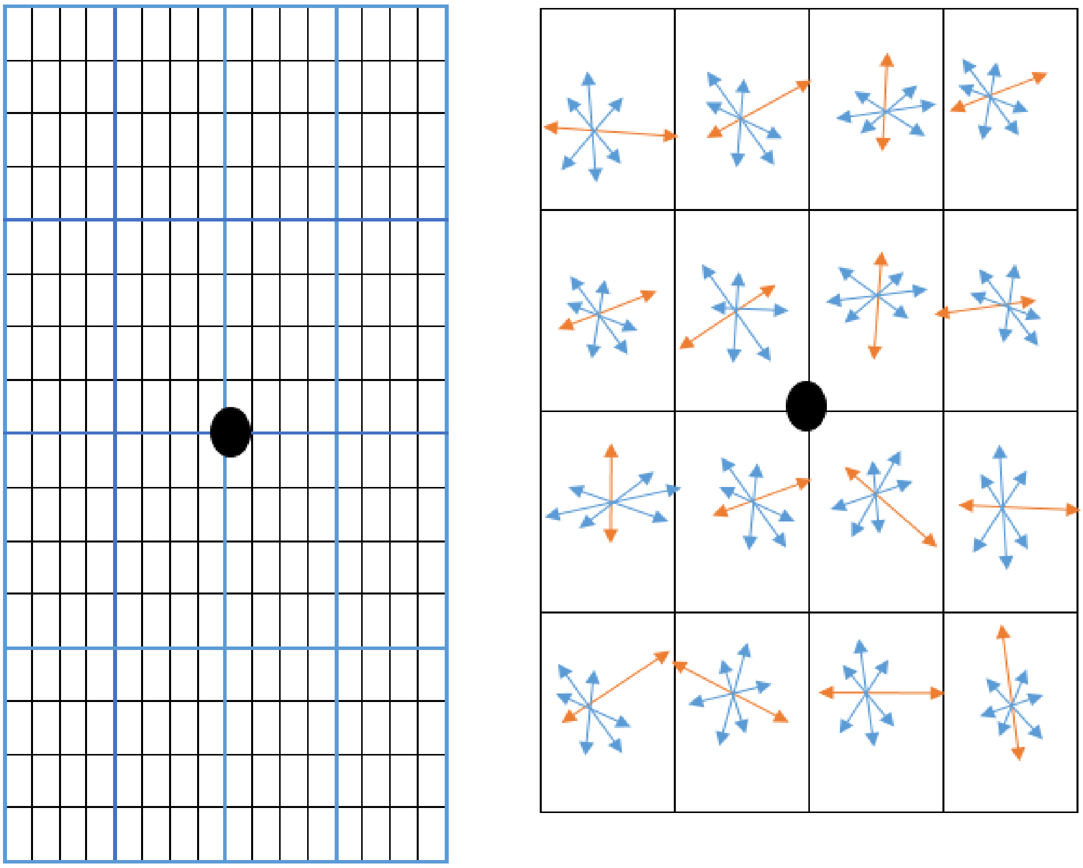


Figure 4.6. 16x16 window (left) and gradient direction per window (right)

Within each 4x4 window, gradient magnitude and orientation are calculated. These orientations are put into an 8-bin histogram.

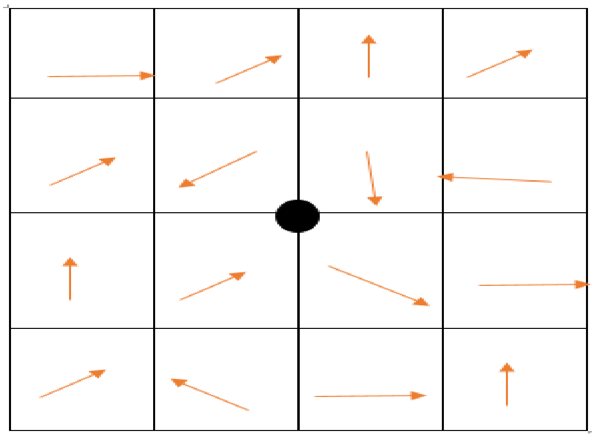


Figure 4.7. Dominant direction selection for a keypoint

The gradient orientation value ranges from 0 to 360. But since the range is divided into ten discrete classes, two values with a maximum difference 44 can reside in the same bin. For instance, all gradient orientation values between ranges 0 and 44, inclusive, belong to the same bin. Likewise, gradient orientation values between 45 and 89 belong to the same bin, second bin. Following the same analogy, the remaining bins are paired with the corresponding gradient orientations. Along with the determination of bins each gradient orientation resides, the amount to be added is also important and it is dependent on the magnitude of the gradient. While constructing the histogram, it is very important to take the distance of each image element whose gradient is being added to the bin into consideration. The gradient of distant image elements contribute less value to the histogram. Mostly, Gaussian function is employed to determine the amount of value added to a histogram. Since there are 16 image elements from each 4x4 windows, multiplying it with the number of bins, which is eight, results to a vector of length 128. Consequently, each keypoint is represented by a 1x128-d vector.

4.4. Machine Learning: Pre-deep Learning Era

Discriminative and descriptive features obtained from digital images convey meaningful information after all are matched into a function through machine learning algorithms. That is, it is nearly impractical to infer a meaningful information from raw feature vectors manually. Therefore, in machine learning, training comprises of a set of specifications that enable algorithms to learn to predict a function ‘f’ that maps a set of features into meaning of multiple forms (such as classes in classification) or to model the underlying hidden structure of data from which a feature is obtained (cluster numbers and so on. As a result, be it be a supervised (classification and regression), unsupervised (clustering and association) or semi-supervised learning, feature vectors are defining, immediate inputs for learning algorithms. In supervised learning, the target is to infer a function or mapping from a sufficient, representative training data that is labelled properly [124]. The training data consists of a number of entities with a set of descriptions called features or attributes. The label refers to a decision or a conclusive meaning to an entity described by the given features. For instance, a set of geometric properties such as height, width and aspect ratio extracted from a binary image of a given character are considered as features. The label eventually is a specific letter that is best described by that particular attribute set. The distinguishing requirement of supervised learning algorithms is supervision. That is, the label (class) representing each entity in the dataset needs to be provided explicitly. This task is mostly handled by human beings and therefore making supervised learning time-intensive and costly. The label can have two or more values given in discrete (classification) or continuous (regression) form. Support vector machines, neural networks, decision trees, Naive Bayes and nearest neighbors are among the most common supervised learning based algorithms.

4.4.1. Support Vector Machines

Support Vector Machines (SVMs), also referred as support vector networks [10], are among the most popular supervised learning algorithms. They are widely used in a range of real time pattern recognition applications including recognition of handwritten digits, scene texts, faces and text language, especially for their high generalization performance and low error rate. In addition to multi-class classification where the task is to assign a given entity into one of the finite number of classes, SVM can also be used to analyze data that requires regression. As a supervised learner, SVM builds a model to predict the class to which an unseen example belongs, only after it is trained on labelled data.

It is also an example of a non-probabilistic, binary linear classifier. A SVM model represents input output pairs in a training dataset as points in space, mapped, so that entities from separate categories are divided by a sufficient, clear margin [124]. Distance between novel examples and the computed gap is used to decide the classes of new examples. Kernel, a name given to the result of projecting data into higher dimension feature space, is one of the fundamental features of SVM. It enables SVM to be competitive in classification tasks where the given data is not linearly separable.

4.4.2. Adversities in classical machine learning

Machine learning, be it supervised or unsupervised, is prone to failure unconditionally for similar applications of varying domains. One of the reasons for such random failure can be attributed to lack of robust feature selection procedures to identify the most descriptive features among the set and reject redundant ones. In general, the most vital challenges in classical machine learning are related to training time, overfitting, computational time and dataset.

4.5. Deep Learning Era

For decades, building a reliable real time machine learning based application was entirely dependent on hand engineered features. Identifying and describing the most discriminative features and useful patterns from data require a lot of expertise and domain knowledge. Even though Neural Networks (NN) are around for long time, it is only recently that the concept of deep learning, NN and optimization frameworks rose to importance and eventually became practical. The primary intention therefore is to automate feature detection and description. It is exciting to see how deep learning is transitioned from a mere, tantalizing concept to a promising, competitive framework that demonstrates an outstanding performance in a variety of areas where classical machine learning algorithms are shown to be unqualified including scene text detection and recognition. Despite the existence of a large number of established studies that prove the previous statement, there is also real obstruction that prohibits deep learning algorithms to be considered as a tool to reach to an ultimate solution. This is mainly because of the thirst of such models for a huge dataset. Existing methods rely on datasets that are obtained from real scenes and synthetic images. However, there are indications that the accuracy of deep learning models is easily taken aback as a result of changes in a pixel or two [125]. There are other drawbacks too. For instance, identifying salient features of an image is impossible. In traditional machine learning, it is not very hard to figure out which feature is carrying the most meaning and which ones are trivial. In addition, even though deep learning models are empowered by high end machines, training in the first place requires far more time than classical machine learning algorithms. Lastly, while synthetic images are included in most datasets to satisfy the need for large dataset, the effect of synthetic noise on the performance of deep learning is not examined. However, it is pointed out that such models are not resilient to artificial noise.

4.5.1. Deep learning components

Neural Networks are the building blocks of modern deep learning architectures. As a result, it is precise to define deep learning as a multilayer stack of NNs, responsible for learning through nonlinear mathematical operations that transform inputs to outputs [126]. Familiar with the traditional feature extraction, deep learning algorithms require a well-defined set of input data which after processing are mapped to an output.

Using the training dataset, the "learning" in deep learning comprises of the adjustment of weights associated with each network in each layer. Adjusted weights for the model are tested to determine if the performance is improved or not. Consequently, training a NN for deep learning is an iterative process. Designing a competent deep learning model requires a set of other vital decisions on activation functions, loss functions, back propagation and hyper parameters.

4.5.2. Why deep learning now?

In general, there are three factors that can be considered as driving forces for rising interests in deep learning. First, hardware: The advancement of fast speed computing devices facilitates various linear and non-linear operations that a NN is responsible for. In addition, searching and optimization through backpropagation are resource intensive operations. With the introductions of GPU (Graphical Processing Unit) and TPU (Tensor Processing Unit), such operations are handled with much more ease and flexibility. Second, deep learning based pattern extraction requires a huge amount of training data to reach to a reasonable prediction or other related tasks. The availability of large scale training data enabled deep learning to be the leading learning technique in image and speech recognition and other intricate scientific and business related applications. Lastly, the availability of tools that are easily adaptable to a specific task also contributed for the increasing popularity. There are a lot of members of different communities that participate in providing, testing and improving resources designed in the context of deep learning.

CHAPTER 5. PROPOSED METHOD : CHARACTER BASED SCENE TEXT DETECTION

Natural scene image text detection, similar to object, face and pedestrian detection starts with a decision on the existence of text embedded in a given image followed by the extraction of the minimum bounding box that encompasses the structural units of text such as characters and words. In this thesis, I designed two different methods for scene text detection. The first detection strategy involves with color clustering using K-means. The second detection is inspired by MSER. However, instead of retrieving stable extremal regions, unstable extremal regions are explored and the resulting technique is referred as Maximally Unstable Extremal Regions (MUER) throughout this thesis. Figure 5.1 gives the general overview of the proposed segmentation and MUER based bottom-up detection and the corresponding recognition strategy. Detailed explanation of the recognition stage is included in the following unit.

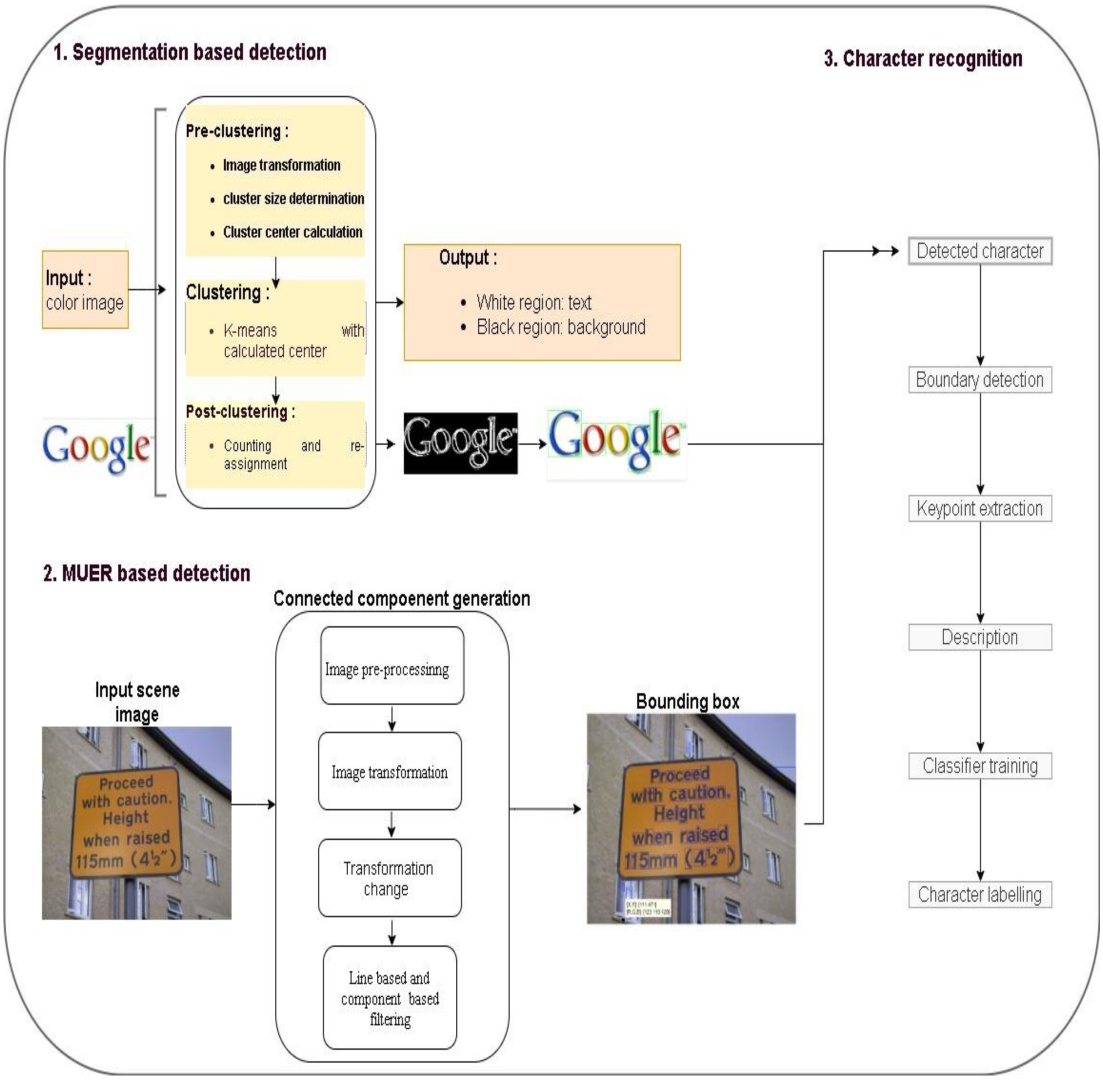


Figure 5. 1. The overall framework of detection and recognition

5.1. K-means Clustering Based Detection (Segmentation Based Detection)

Color clustering using K-means has been applied for scene text segmentation. In this thesis, differing from existing techniques that benefit from the same principle, emphasis is given on the interpretation of the clustering results and the number of clusters. The detection method includes three important phases. First, the input image, multi-color scene image is converted to its HSV equivalent, which is used as a reference to set the number of clusters. Moreover, the original input image of size specified with height and width is represented with three column vectors of size 'width*height' each signifying the red, green and blue color components respectively.

The second phase is clustering. However, in K-means, the number of clusters is required to be specified explicitly. Therefore, a simple test, image contrast test, is followed. This is done according to the sum of the standard deviations in H, S and V. If this sum is less than 0.5, the image is said to have low contrast and the number of clusters is decided to be two. Otherwise, the clustering size is determined to be three. Algorithm 1 summarizes the operations in this section.

Algorithm 1 Clustering colour channels

```
Input:  RGB values of a colour image.
Output: clusteredMatrix [a matrix of rows and columns of the same size with
 the original image and cluster number entries]
1: Read a coloured image
2: Color conversion: from RGB→HSV
3: Compute $\sigma$[H], $\sigma$[S] and $\sigma$[V]; standard deviations in all converted color
channels
4: Add  results of step 3, store into a temporary variable std.
      std= $\sigma$[H]+ $\sigma$[S]+ $\sigma$[V],
5:  Compare if the result from earlier step is greater than a threshold
if [std > 0.5] then
 Set ClusterSize to 3
else
set ClusterSize to 2
 6: end if
 7: k-means [RGB, ClusterSize]
 8: Store the cluster numbers
```

Since the input image is uncropped scene image, mostly texts on these images are intended to provide clear information such as titles, names, locations and direction. As a result they usually occupy the least portion of the image. That is the total number of image elements that form text regions are far less than the remaining pixels forming the background. Therefore, the next phase is to count the number of pixels in each cluster. A cluster with the highest number of image elements is assumed to represent background elements. Therefore locations with an entry that symbolizes the cluster with the most image elements is set to zero. Likewise, the cluster number with the minimum number of elements is assumed to represent text regions and eventually are set to one. If the image is of high contrast and therefore the elements are clustered into three, the second cluster which has total elements between the maximum and minimum, requires further consideration. Consequently, if the elements in such clusters comprise more than one third of the total number of pixels, then the elements in this cluster are also assigned to the background. The following pseudocode outlines the basic procedures in assigning pixel elements to one and zero.

Algorithm 2 Cluster size determination

```
Input:  Matrix with cluster number
Output: Binary matrix
1: Initialize an array BinaryArray of size width*height
widthCount =1
 heightCount =1
2: count entries of the resultant matrix from Algorithm one.
    totalOne= sum[clusternumber=1]
totalTwo= sum[clusternumber=2]
totalThree= sum[clusternumber=3]
3: Do while widthCount < width
4:Do while heightCount < height
5:If clustered[widthCount,heightCount]=1and totalOne>
[widthCount*heightCount]/3 then
         BinaryArray[widthCount, heightCount] =1
else
         BinaryArray [widthCount, heightCount] =0
 6: elseif clustered[widthCount,heightCount]=2 and totalNumberofClusterOne
> [widthCount*heightCount]/3  then
binaryArray[widthCount, heightCount] =1
else
         binaryArray [widthCount, heightCount] =0
```

```
end if
    EndDo
```

One of the evitable disadvantages in all detection techniques except for deep learning based ones is post-processing. That is, there are situations where non-text pixels are returned as text and vice-versa. A detection technique should guarantee that the returned text region is composed of only text pixels. In this thesis, a filtering technique special to the detection method explained in this subsection is explained with the help of the following algorithm.

Algorithm 3 Filtering false positives

```
Input: Resultant binary image from Algorithm 2
Output: Filtered binary image where false background pixels are restored to foreground
Count consecutive ones in each row of the binary image
Identify the row with the longest white image region.
if length (whiteRegion) is greater than 150 then
Identify the cluster to which these white pixels belong to
Iterate through the entire binary image, replace the binary value '1' with '0'.
```

5.2. Maximally Unstable Extremal Regions Based Detection (MUER)

In the method discussed below, the image is assumed to be acquired with the text in mind; text being the center of attention. In other words, the study is categorized under focused scene text detection. The difference is, if text is incidental, text can appear in any direction and with many distortions as a result of occlusions and clutter. On the other hand, if the text is focused, the text is mostly in horizontal or vertical direction. Moreover, physical obstacles are avoided during the time of image acquisition. Before the actual task of detection, a set of pre-processing operations are employed on the input images. The general framework for the detection stage is given under Figure 5.2.

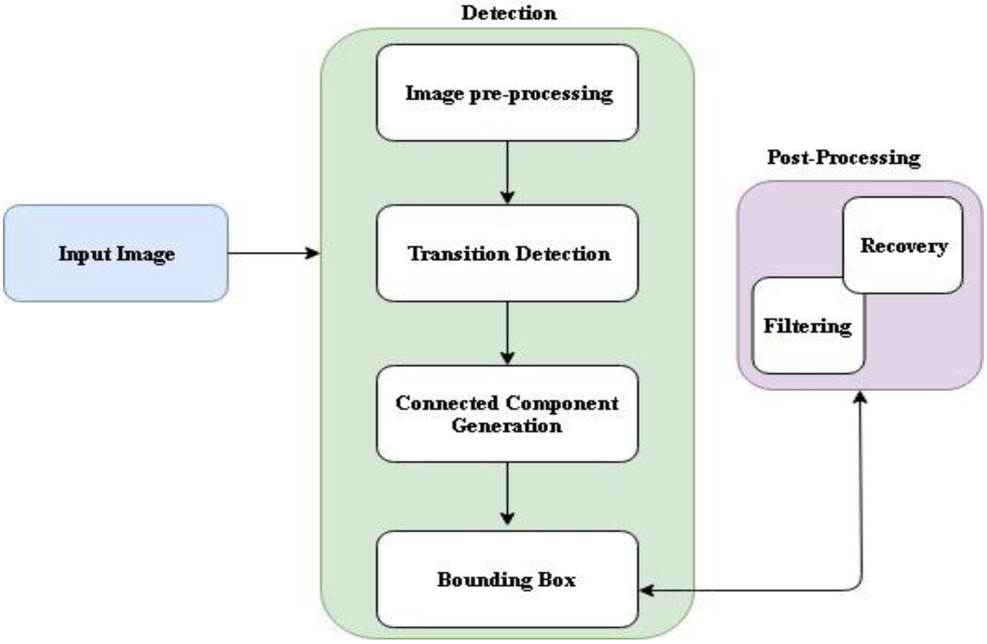


Figure 5. 2. MUER based Scene character detection framework

5.1.1. Image pre-processing

The pre-processing stage is primarily aimed at converting the color scene image into a form that is easy and convenient to mark changes in image intensity. There are a number of functions that can be used for this purpose. In this thesis, the reciprocal of R, G and B values are added to give a single result as given in Equation 5.1. This stage is fundamentally equivalent to other methods that rely on gray scale conversion. One of the reasons that Idevised to use such representation is to avoid gray scale conversion, which as is known, has a limited range [0-255]. This procedure helps preserve as much information as possible and distinguish individual elements considering this value and the spatial relationships.

$$RGBtoIntermediate = \frac{1}{r} + \frac{1}{g} + \frac{1}{b} \tag{5.1}$$

Where the r, g and b represent the red, green and blue channel components of the original image.

The resulting matrix has entries between 0.0039 (maximum R, G and B) and 3 (minimum R, G and B).

5.1.2. Image transformation

In some computer vision applications where color intensity gives very little information, spatial relationships between image elements can be considered as a rich source of information. Scene text detection and recognition is one among such applications. Spatial relationships can be expressed in terms of distance, direction and area. For this particular case, Iconsider distance between points that are topologically closer.

For each image element located at position (x, y), the sum of horizontal and vertical distances from a fixed location (0, 0) on a grid (Manhattan distance) is computed. The result obtained from this operation is transformed into an intermediate value with the help of two functions: Logarithm and square root function. Re-expressing values as square roots, logarithms, or reciprocals, for example, can often facilitate interpretation by simplifying the appearance of data [127]. The Logarithm function can be used to compute an absolute relative change in intensity between two neighboring pixels.

The decision on which function best fits a particular input depends on the image. For input images which exhibit relatively high contrast, square root function is applied. Otherwise, Logarithm function is preferred as it is easier to track small changes in foreground and background.

5.1.3. Image contrast and function selection

In order to be able to choose one of the two functions, first the image is analyzed to determine if it is high contrast or otherwise. As a primary step, the HSV values are derived from the corresponding RGB values. If the mean of the light intensity (V) is less than a given threshold, the image is said to exhibit low tonal contrast. As a result, such images are re-expressed with square root function. On the other hand, if the value obtained is greater than a threshold, the image is said to be highly contrasted and therefore are transformed with logarithm function. After a series of experimentation, the threshold is determined to be 0.5.

The rate of the transformed value to the entries obtained from Equation 5.1 is the final output. Rapid intensity changes marking transitions from background to foreground and the reverse are retrieved based on the results of Equation 5.2 for low contrast images or 5.3 for relatıvely high contrast images.

$$Transformed(x, y) = \log(x + y) / RGBtoIntermediate(x, y), \tag{5.2}$$

$$Transformed(x, y) = \sqrt{(x + y)}/RGBtoIntermediate(x, y) \tag{5.3}$$

Where *(x, y)* specifies the horizontal and vertical axes of a given image element and *(0, 0)* is a fixed location considered as the origin.

5.2.4. Locating high and low transitions

Considering the transformed value, for any two successive points located at (x, y) and (x, y+1), the difference between the given locations is computed with Equation 5.3. This process is reiterated for all image elements by sampling two points at a time. This step is fundamental to locate an atypical rise or drop in intensity.

$$change(x, r) = Transformed(x, y+1) - Transformed(x, y) \tag{5.4}$$

Computing the difference is followed by transition point detection. In the following sub-sections, a transition that marks an intensity change from low to high is termed as high transition (*change* has values greater than a threshold) and the reverse is referred as low transition (change has value less than a threshold).

Next, using the results obtained from this stage, both high and low transition points are marked as white. Matrix entries which are greater than 2 or less than -2 are considered as high and low transition points respectively. These thresholds are set according to the following facts.

a. If two neighboring pixels (x, y) and (x, y+1) have minimum value (0.0039) from RGB transformation, then the difference is zero.
b. If two neighboring pixels (x, y) and (x, y+1) have maximum value (3) in RGB transformation, then the difference is zero.
c. If two neighboring pixels (x, y) and (x, y+1) have different values, and the first has minimum value and the second has maximum value, then the difference is greater than two. Consider these pixels as transition points.

In addition to pixels which are located next to each other (left and right), pixels which are at the top and down of a given pixel are also considered.

Sample output from this stage is shown in Figure 5.3 as a binary image where all the high and low transitions are set to one and the remaining positions are set to zero.

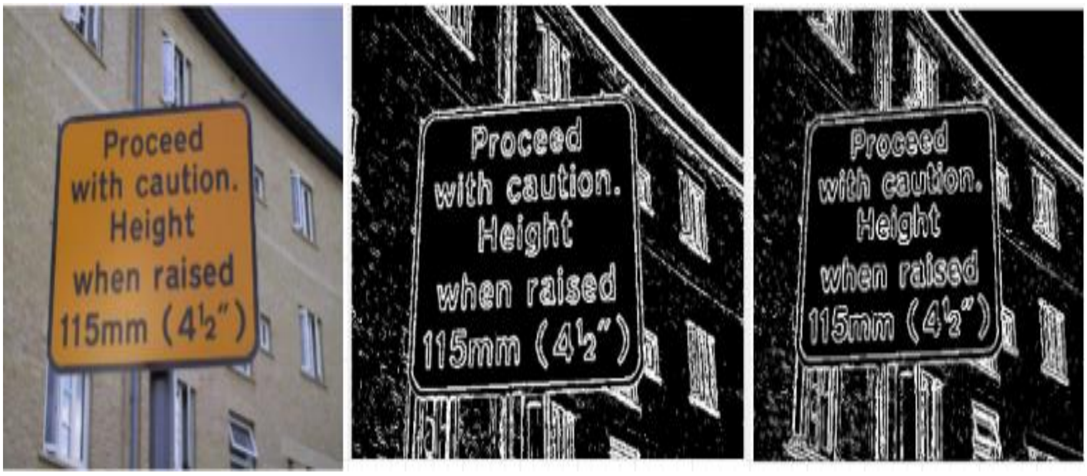


Figure 5.3. Original, high transition and low transition images respectively

The high and low transition based binary images are combined with Logical OR to form a connected component representing mostly a character. In cases where the characters are separated with a relatively low space, the connected component represents the word formed from the nearest characters.

5.2.5. Detection and bounding boxes

Results obtained from the previous sub-section are binary images where white pixels represent both high and low transition locations and black pixels represent regions with relatively steady intensity.

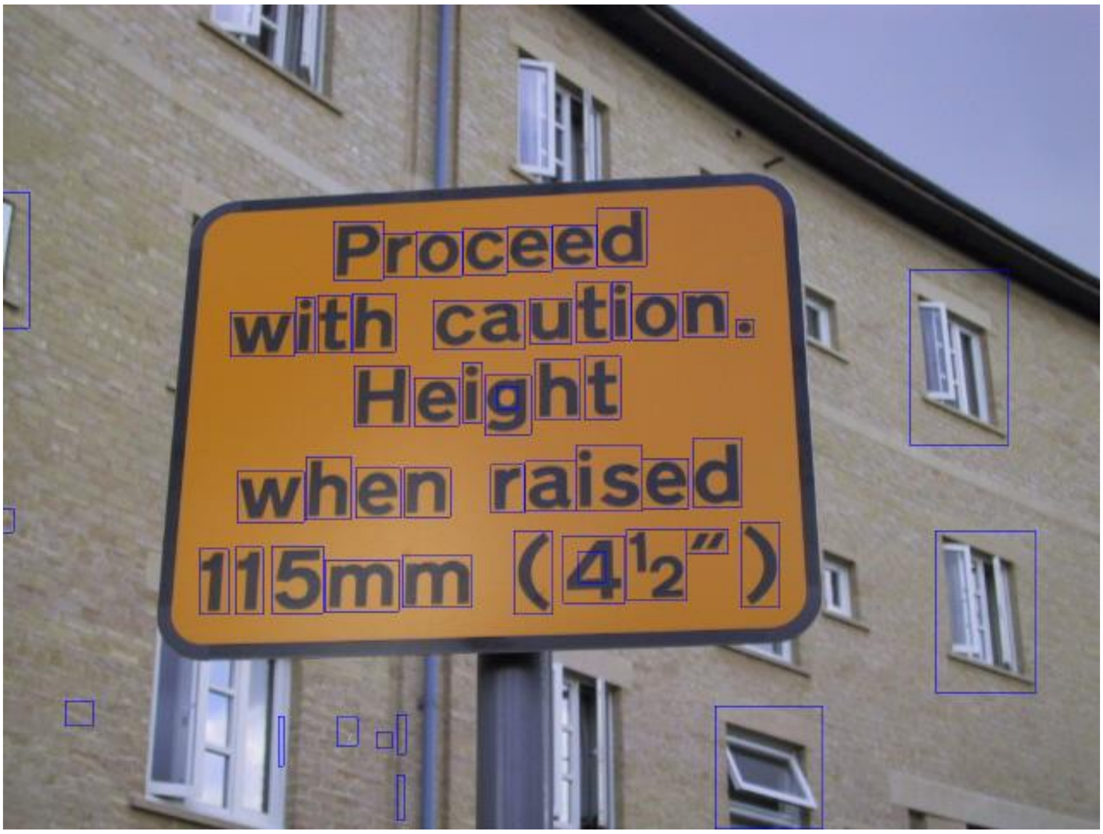


Figure 5.4. Bounding boxes around candidate characters on the input scene image

As shown in Figure 5.4, most returned bounding boxes represent characters, some non-character regions and merged characters as well. As a result of merged characters, the detection can be counted as unsuccessful or merged characters can be rejected as false positives. In such cases, the precision of the algorithm will be relatively low. In order to resolve this issue, post-detection procedures called filtering and recovery are included.

5.2.6. Filtering

The filtering stage is common to all detection tasks where certain image features are set to decide whether a given region belongs to an accepted class or not. A successful filtering technique improves the precision of a given algorithm considerably. The goal is to enable the algorithm re-label false text pixels back to background pixels.

As a primary coarse-level filtering, a text-line testing based filtering is designed. Positive text lines are retrieved by considering standard deviations computed from the intermediate image gradient which is referred as Transformed in section 5.1.3. Now, considering the binary image, for all image locations whose corresponding entries are one, the gradient magnitude is retrieved and stored. Similarly, locations with zero entries in the transformed value are set unchanged. Next, the standard deviation is computed row by row to determine the image rows with the highest and lowest gradient change. K-means clustering with a calculated center is employed to cluster all the image rows into two, supposedly suggesting text and non-text rows. To determine the text regions from a set of rows, the following heuristic rules are applied.

a. A set of rows is accepted as a text region if it includes a minimum of ten consecutive rows.
b. If all image rows are clustered into the first group (high standard deviation), the text region is assumed to range from the first to the last row.
c. If all the text regions have less number of rows and as a result the algorithm returns all rows as a background, merge all rows of high standard deviation and return it as a text region.

The procedures followed during filtering are summarized with the following algorithm.

```
Algorithm Filtering false positives
    Input: Binary images of low and high transitions
    Output: passed text lines
```

```
1: calculate gradient from the transformed RGB value
2: store gradient values at locations where the binary image is one.
3: calculate the standard deviation of gradients at locations from step 2.
4: Find the minimum and maximum standard deviation.
5: Cluster the text lines with the two values as centers.
```

As a fine-level filtering, Iemployed component level filtering where each connected component is tested to be accepted (text) or rejected (background) connected component.

Algorithm 3 Filtering false positives

```
Input: Connected Components
Output: Filtered binary image where false text pixels are restored to
background
1: With the bounding box bounding the connected component, extract the
   corresponding region from the transition image (high transitions OR low
   transitions).
2: Count the connected components in the extracted region.
3: if length (connectedComponents) is greater than 40 then
4: set all the image elements of current connected component to 0.
5: Iterate through the entire connected components, replace the binary value
   '1' with '0'.
```

5.2.7. Recovery

The goal of recovery is the reverse of filtering. The recall of a given detection algorithm determines whether all objects/text regions whose ground truths are provided by an annotator are detected or not. Recovery improves the recall of a given detection algorithm as false background pixels are re-labeled to text pixels.

Conclusion

Scene text detection is studied and is addressed only partially. There are still gaps that require immediate and serious consideration. Even though content complexities such as inter-word color, size and language variance are covered in detail, there are unresolved issues including intra-word color differences, random orientation and multi-language. The designed techniques address these issues to some degree. For instance, both detection techniques are able to detect texts of multi-orientation and language. Besides, with the first detection approach, a single word that is composed of multi- colored characters is also detected. However, there are limitations prime to both detection strategies. The most crucial one is lack of discrete principles during post-processing. Both filtering and recovery are involved with geometric feature centric heuristic rules. The other one is prominent with the MUER based detection. Since pixel differences are computed within a loop, it is relatively slower. Post-processing is a problem as well. Curvature information is not enough to distinguish character regions from non-character region.

CHAPTER 6. PROPOSED METHOD :CHARACTER BASED SCENE TEXT RECOGNITION

In the previous chapter, I have explained the proposed scene character detection approaches along with their inherent advantages, novelty and weak sides, based on performance reports generated from comparisons with state-of the-art methods. Scene text recognition stage can be seen as an extension of the detection stage in a sense that they are two highly interconnected procedures. Recognition is successful after successful detection and detection is meaningful only after successful recognition.

In classical scene text recognition, neither character, nor word based scene text recognition are single phase solutions. Rather, a serious of activities are involved, each responsible for a specific output. In this thesis the recognition process starts with a segmented character image as an input. Fundamental sub-tasks include image pre-processing, boundary detection, curvature calculation, key point identification, feature description and SVM training. A general outline of the system that summarizes scene character recognition method introduced in this thesis is presented in Figure 6.1.

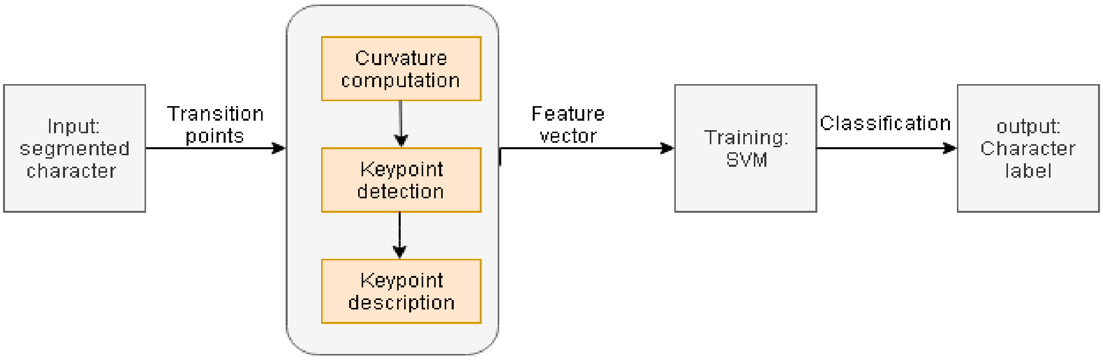


Figure 6.1. Scene character recognition framework

6.1. Basic Operations

Labelling characters in character recognition based scene text recognition techniques is the goal behind training a supervised learning based algorithm such as SVM. This is accomplished with the help of description (attributes) and decision (classes) associated to each example in the training dataset. Even though training requires relatively shorter time, identifying the best attributes is very tedious. Only with descriptive attributes will the learner be able to do its best on unseen examples (test images). Therefore, it is always recommended to assign a significant amount of time to attribute identification (feature detection and description). The following sub-sections explain critical operations that directly influence the results obtained from classifiers.

6.1.1. Pre-processing and boundary detection

Pre-processing, as the name suggests, is the process of preparing images in terms of size, intensity, color and contrast. It is such a fundamental factor that mostly improves the image appearance which in turn affects the effectiveness of other succeeding operations. Therefore, it is unavoidable in almost every image classification and pattern recognition application. Being not an exception, in scene text recognition, especially in the designed technique, transformations such as RGB to binary, size transformation (size normalization) and preparation of the cropped character for transition point detection are carried out as a pre-processing. The attributes describing a character image are fetched from the boundary of the binary image. Therefore, boundary detection is considered as the next, prominent input for the keypoint detection. However, for natural scene images of uncropped, non-monochrome words and characters, this operation is included at the very beginning of detection.

6.1.2. Keypoint detection

Keypoints are distinct image elements that usually signify transitions in color, intensity or contrast. In this thesis, the keypoints are pixels that cause a significant shape change in the image boundary or a connected curve. That is, the contour or boundary of the character image is assumed to represent a continuous curve. With this assumption, curvature information for each point on the boundary of the image is sought. In this case, instead of the exact boundary, the transition points (both high and low) during the preliminary stage of detection are considered as an approximation for the boundary. Iterating through all the transition points, a value that hints how much an input image boundary is concave or convex at a particular transition point is calculated. The result, which may be negative, positive, or zero suggests concave, convex, and straight-line transitions respectively. All transition points with curvature that is greater or less than zero (concave and convex transitions) are candidate keypoints. Final keypoints selection is dependent on simple comparison of the remaining concave and convex transitions with a threshold. The threshold for this stage is set according to a series of experimental analysis.

6.1.3. Curvature calculation

Instead of other geometric metrics that are widely used in image shape representation in the literature, in this thesis, curvature information is employed mainly because of the following two reasons. First, since characters are distinct objects, shape information is regarded as the best representation for such objects and curvature provides the most information about shape than other metrics. Second, curvature information is independent of local coordinate frames [128]. That is, plane transformations such as rotation and translation hardly cause changes in the computed curvature value.

Curvature can be calculated in various ways. Some of them are methods using finite differences, geometric relationships, and moving frames [129] .Regardless of the invariance property of curvature, methods that are used to compute the value however are sensitive to some extent towards certain transformations. Therefore, a geometric technique which is invariant to affine transformation, especially rotation, is given priority over other techniques. Moreover, geometric technique does not necessitate equality of spatial distances between points. A brief explanation on the computation of curvature in the context of a random osculating circle is included in the following section.

The radius of a random osculating circle matching three points is computed as the ratio of four times the area of a triangle formed from the given points and the product of the distances between each pair of points forming the triangle. The reciprocal of the radius of the osculating circle is considered as the curvature of the middle point, where three neighboring points are used in each iteration. Here, the assumption is to approximate the image as a continuous curve, and the extension of the boundary along the X-axis approximates the length of the curve. Therefore, at each time, three successive points A (x_1, y_1), B (x_2, y_2) and C (x_3, y_3) are examined along the boundary and the curvature of the second point in a given iteration is computed as given in Equation 6.1.

$$curv = 1/r = 4(area(\Delta ABC))/abc \tag{6.1}$$

Where r is the radius of an osculating circle, ΔABC is the area of a triangle formed from the corresponding three points, and $a = (x_2 - x_1, y_2 - y_1)$, $b = (x_3 - x_2, y_3 - y_2$ and $c = (x_3 - x_1, y_3 - y_1)$ denote lengths of the sides of triangle ABC.

At any given iteration, three succeeding points are considered. The resulting curvature value obtained as a result of computation indicates the concavity or convexity of the boundary, particularly at the second point. This value is equivalent to the inverse of the radius of any given circle that passes through these three boundary points. The triangle formed from any three points can be described by attributes such as sides, area or perimeter. Among these, area and side length are relevant to calculate curvature and determine how much concave or convex a given transition point is.

6.1.4. Keypoint selection

Usually, a large number of boundary points exhibit a non-zero curvature. In order to identify the most distinct, descriptive boundary points, and ultimately avoid wasting resources in unnecessary description, a comparison operator is employed to identify a set of boundary points satisfying the minimum requirement. Calculating the curvature value for each boundary pixel is followed by a selection procedure where a pixel's curvature is compared to a certain threshold value. In fact, different thresholds are applied to find the optimal keypoints. However, increasing the threshold causes loss of important features especially in cases where the input is low contrast image. On the other hand, decreasing the threshold enables most of the pixels with non-zero curvature value to be accepted as keypoints, which, in turn, leads to poor recognition. In this study, after a series of experiments on different character images, 0.3 is determined to be the most favorable curvature threshold.

6.1.5. keypoint description

The results obtained from the keypoint detection stage are not in a suitable format if they are intended to be used directly. For use along with a specific learning algorithm, feature detection needs to be followed by an equally important phase: feature description, where characteristics of image elements around the detected keypoints are transformed into a discrete representation. In this thesis, physical properties such as location, angle measure and direction of gradient obtained from the image are primarily considered for the description. In addition to the detected keypoints, physical locations of transition and center points are employed as well. The final result of description, a vector, has a dimension which is equivalent to the sum of the lengths of the constituent attributes. These are the only direct input values to any supervised learning based classification, clustering or regression tasks. A given feature vector presents a special meaning to an end user after mapped into outputs such as classes and clusters through a specific learning algorithm.

6.1.6. Distance calculation

Euclidean distance between a pair of points taken from keypoints, boundary points, and individual point pairs from both sets and image center is computed. Moreover, the angle between these point pairs are also considered as an imperative description of keypoints and the points around the keypoints. Therefore, four important feature vectors are extracted as follows. In all cases, values obtained from a given computation are represented in histograms.

a. Maximum distance (MD) between a pair of points from keypoint set: For each keypoint in the set, the maximum distance from all the distances computed with the remaining keypoints is stored.

b. Distance between a pair of points taken from keypoint set and boundary set (DKeyBound): For all points in the keypoint set, its distance from the corresponding boundary point is computed.

c. Distance between boundary points (DBound): a pair of points in the same set, boundary set, are considered at a time and the distance between such points is computed.

d. Distance between keypoints and center (DCenter): a pair of points from keypoint set and image center are considered and the distance between these two points is calculated for all keypoints.

Distance computation is followed by histogram based representation. All the four values from obtained from the previous computations are represented in histogram of varying bin numbers. The number of bins is determined based on a series of experimental results. Therefore, the dimension of each feature vector is specified as 64, 35, 35 and 32 respectively. Figure 6.1 illustrates distance between a pair of points taken from keypoints, boundary points and image centre respectively.

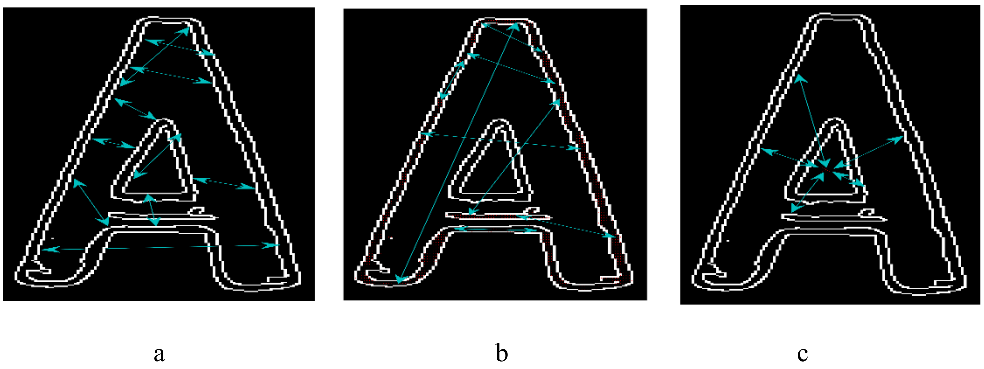


Figure 6.1. Distances between boundary points (a), keypoints (b) and image centre (c)

6.1.7. Image gradient direction calculation

In addition to physical distances between various types of pixels, another parameter, image gradient direction, is also used to represent characteristics of points surrounding keypoints. This is used to get information about the direction at which the image intensity is changing more drastically. The result of this stage is a set of feature vectors of different dimensions. Both grey and binary image equivalents of the original input image is used to derive the gradient direction at the boundary and keypoints. These values are referred as GGD and BGD from now onwards to imply gradient direction of boundary and keypoints on the grey and binary images respectively. Figure 6.2 illustrates gradient direction of a grey image.

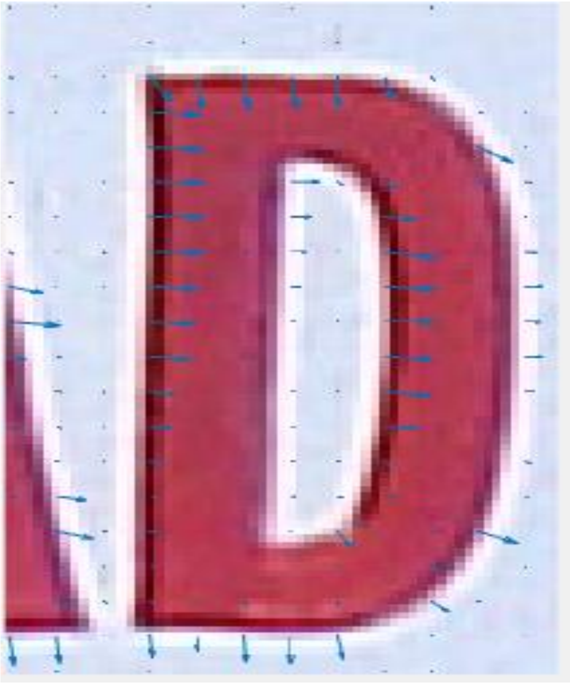


Figure 6.2. Gradient direction: blue arrows

Considering any point (x, y) on the input image (grey or binary), gradient direction is computed using Equation 6.2.

$$\theta = \tan^{-1} \frac{\mathrm{Im}(x, y+1) - \mathrm{Im}(x, y-1)}{\mathrm{Im}(x+1, y) - \mathrm{Im}(x-1, y)} \tag{6.2}$$

6.1.8. Determination of an angle between a pair of points

The last parameter that is considered to form the final feature vector is an angle between a pair of consecutive points (boundary, key points) with respect to the horizontal. This value is computed for all points of interest with Equation 6.3.

$$\theta = \cos^{-1}(x_2 - x_1) / d \tag{6.3}$$

Where x_1 and *x2* are the x coordinates of two consecutive points and d is the Euclidean distance separating the two points. Similar to the earlier description parameters, in this case too, the value is represented with a histogram of 64-bins from each type of points (keypoints and boundary points). The final feature vector is obtained by simple concatenation and it has 128 dimension.

Lastly, all the feature vectors obtained from a number of parameters are simply merged to describe the keypoints of a given image and as a discriminative feature for a particular character image. The sum of the individual feature vectors is the dimension of the final description. As a result, the designed feature description 573-dimension. Irefer the designed feature as Global Curvature Feature (GCF) in the following sections.

6.2. SIFT and GCF

There are a number of characteristics to be measured in order to determine the robustness of a feature detection algorithm. The primary one is affine invariance. There are a number of successful detectors suggested in the literature that are claimed to be successful achieving a high-level of invariance when it comes to scale and rotation. One of such detectors is SIFT.

In this thesis, the performance of GCF in cropped character recognition is compared to the results of recognition with SIFT because of the detector's resilience to scale and rotation. Moreover, unlike other state-of-the-art region detectors such as MSER, both methods are based on keypoint detection. As a result, it is reasonable to compare and contrast recognition accuracies for the task required.

In addition to SIFT, Ialso examined the performances of other feature detector, HOG, which is initially proposed for a different purpose than character or text recognition. Although HOG has a higher feature length and low prediction speed than SIFT and the designed feature detector, the results of classification suggest that it is effective in terms of classification accuracy.

The experiments are conducted using MATLAB R2016a on a PC with Intel i5 CPU, 8GB RAM. All the results reported in this section are carried out on the most widely used standard scene character datasets; Chars74k and ICDAR 2003 robust character recognition datasets.

6.2.1. Chars74k dataset

In the first setting, 1068 cropped images of characters from the Chars7k dataset are chosen randomly. Both SIFT and GCF feature sets are extracted from the sampled images. Example images demonstrating keypoints detected with the two methods are given in Figure 6.3.

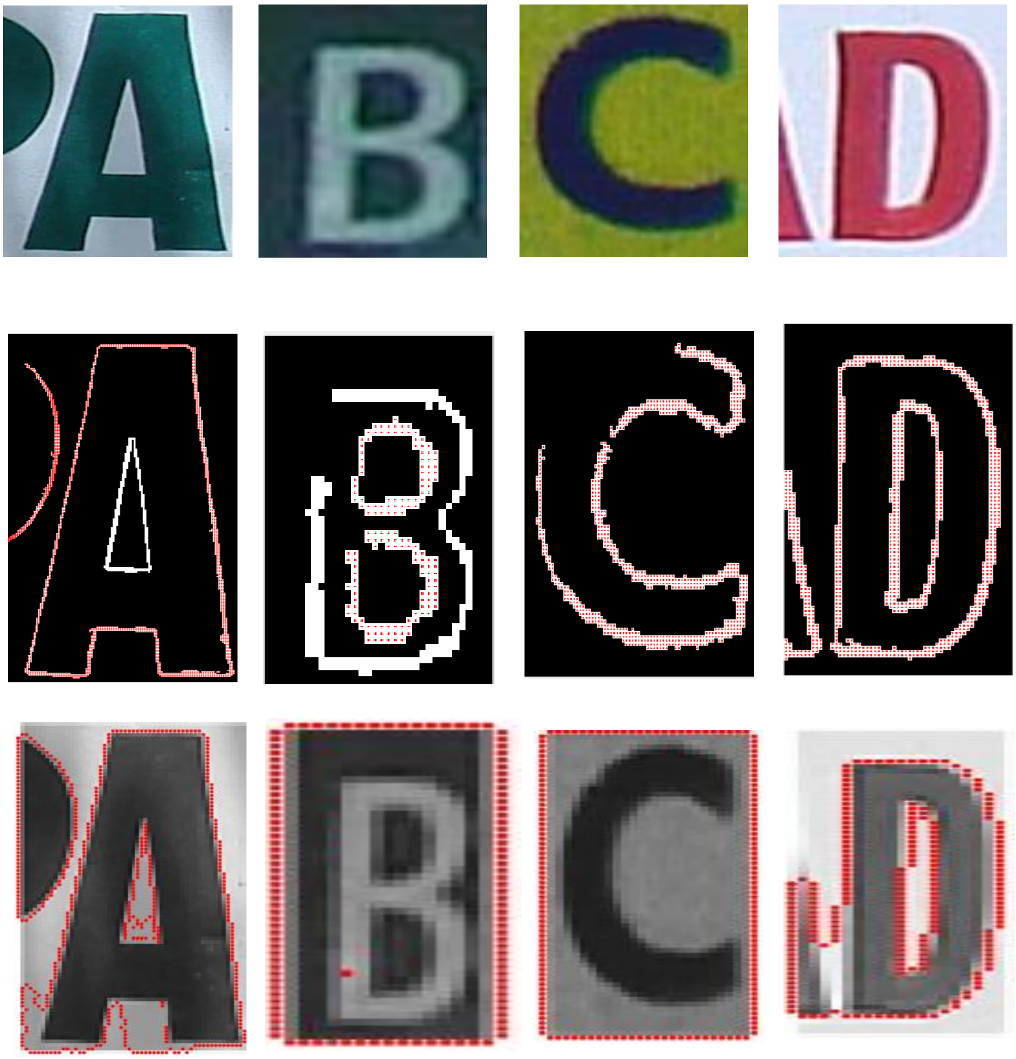


Figure 6.3. Keypoint detection: original images (first row), designed method (second row) and SIFT

Besides its computational complexity, it is clear that SIFT features are not reliable for images of certain properties such as blur and uneven illumination. (Refer Figure 6.3, third row, alphabets ‘B’ and ‘C’). Contrarily, the designed method detects a reasonably sufficient number of high and low transitions. As a result, there is a higher possibility to locate representative keypoints than SIFT. While qualitative measurement can be used to determine the effectiveness of the detector, the descriptor’s effectiveness is measured quantitatively, mostly with the classification accuracy of a particular classifier. To measure the effectiveness of the designed feature, the classification accuracy of SVM is given in brief in chapter 7.

While a theoretical background and deeper explanation on SVM is included in Section 4.3.1, in this section, first, justifications on the basic reasons for its wide use in computer vision applications such as handwritten text recognition, object recognition and born digital image text recognition are highlighted. Lastly, classification accuracies of SVM with different kernels trained on both SIFT and GCF is provided.

Although there are several successful classifiers such as neural networks, in this thesis, SVM is the primary choice because of the following reasons.

a. SVMs are reported to demonstrate high classification accuracy in medical diagnostics, OCR , electric load forecasting and other related fields [130].

b. With the help of kernels (Quadratic Linear and Cubic), SVM can act as a non-linear, non-parametric classifier and hence can be used for multi-class classification problems.

c. SVMs are known to be robust regardless of biases exhibited by a training dataset.

d. SVMs provide exactly one solution and therefore are robust over different samples.

CHAPTER 7. EXPERIMENTAL RESULTS

In this chapter, experimental results of the two scene character detection approaches and scene character recognition method introduced in this thesis are presented. Correspondingly, the resulting comparisons and limitations seen in the designed methods are discussed.

7.1. Datasets and Performance Reports: Segmentation and MEUR Based

The detection methods are evaluated on two types of datasets. The first, clustering based segmentation is tested on multi-color scene images crawled from google images. The MUER based method is tested on images from ICDAR 2013 born digital image dataset. On the other hand, the accuracy is measured quantitatively with OCR recognition accuracy (ABBY FineReader and Google's OCR) and the number of intersection between the detection and ground truth respectively. Sample outputs from clustering based multi-color scene image segmentation are also included in Figure 7.1.

The performance of segmentation based scene text detection methods is usually determined with the number of characters and words recognized correctly after binarization. The same number is determined before binarization too. The difference between these two values signifies the performance of a given binarization algorithm. The word and character recognition accuracy of Google's OCR employed on images taken from Google images both before [word recognition before binarization (WRBB), character recognition before binarization (CRBB) and after binarization [word recognition after binarization (WRAB), character recognition after binarization (CRAB) is depicted in Table 1.

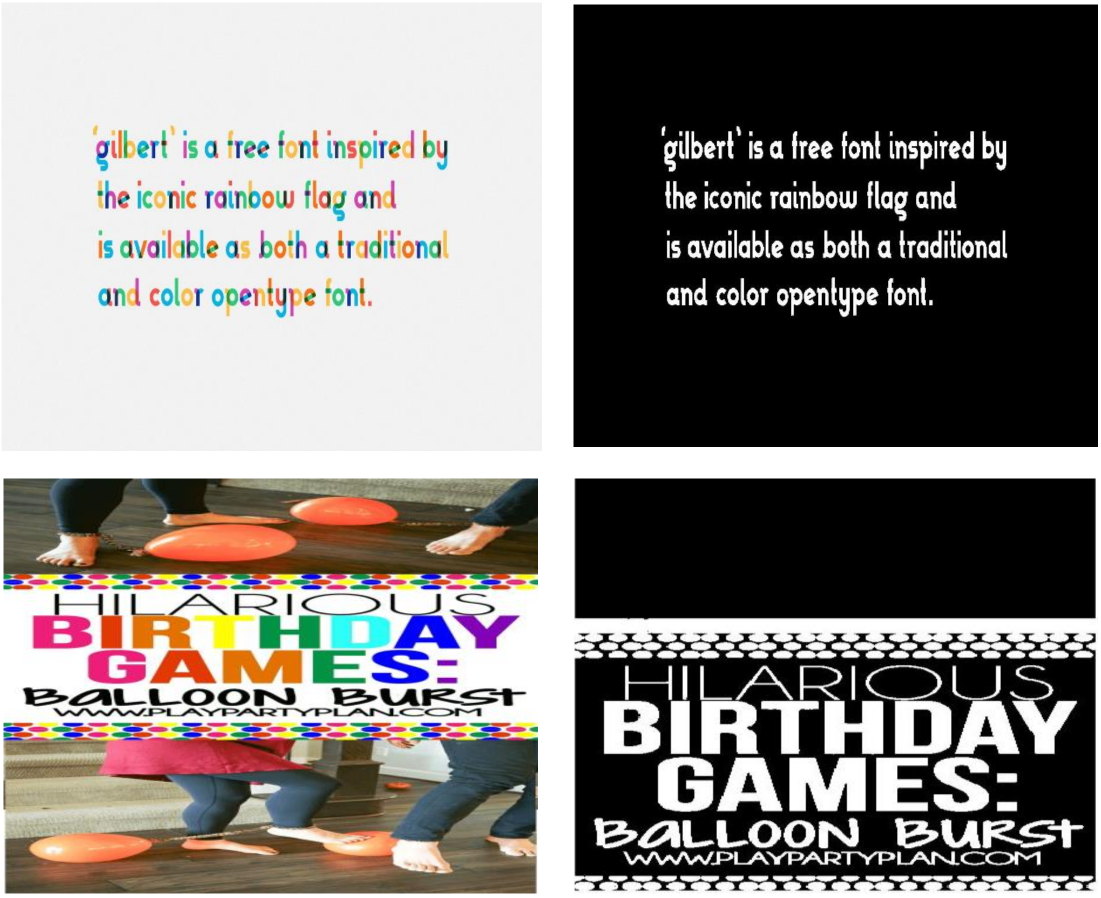

'gilbert' is a free font inspired by
the iconic rainbow flag and
is available as both a traditional
and color opentype font.
HILARIOUS
BIRTHDAY
GAMES:
BALLOON BURST
WWW.PLAYPARTYPLAN.COM

Figure 7.1. Multi-color scene image segmentation, left: original and right: binary results

Table 7.1. OCR accuracy of words and characters from Google images.

WRBB (%)	WRAB (%)	CRBB (%)	CRAB (%)
34.78	43.7	48.29	59.18

Similarly, for ICDAR 2003 robust reading dataset, ABBY FineReader was employed to determine the performance in terms of characters and words recognized correctly as before. Table 7.2 summarizes the results.

Table 7.2. OCR word accuracy from ICDAR 2003 dataset

WRBB (%)	WRAB (%)	CRBB (%)	CRAB (%)
71	74	74	77

In conclusion, binarization is a pre-processing step whose success has a great effect on other succeeding tasks such as recognition. Due to the presence of diverse complexities such as illumination, blur, perspective distortion and camera based problems; scene text binarization is relatively complex than document image binarization. Researchers from various communities have examined the topic from multiple perspectives. Despite the efforts, it is apparent from the literature that no binarization technique works best for all types of scene images. In addition to common problems in natural scene image, colour variance among characters of the same string is particularly dealt in this subsection. Relying on the proportion of text region to the rest of the image, image elements are grouped into clusters through K-means clustering algorithm. The binarization step depends merely on the number of image elements in each cluster. Each pixel location is updated with the value of the total number of elements, which belong to the same cluster as that specific image element. This number determines if a given image element forms an image background or text.

The second detection which is based on the detection of MUER as character candidates is tested on ICDAR 2013 born digital image dataset. The performance is measured on pixel wise accuracy against the ground truth. The precision, recall and F-score is given in the following table.

Table 7.3. Pixel level accuracy (methods in ICDAR 2011 robust reading competition, proposed method)

Method	Pixel level accuracy (%)		
	recall	precision	F-score
Proposed ((MUER	61	88	71
Anthi-mopoulos	86.15	71.19	77.96
OTCYMIST	80.99	71.13	75.74
TextTorter	65.2	62.5	63.82
SASA	71.93	54.78	62.19

The pixel-wise performance measure gives an exact performance measure, where a single image element detected or missed is counted. This protocol avoids most of the inconveniences that are peculiar to current scene text detection performance evaluation metrics which either excessively punishes or rewards a method. However, it is relatively time consuming as pixels and not regions, are units of measurement. As is shown in Table 7.3, the proposed method has high pixel-level recall than the rest of the methods. However, the pixel level precision is relatively low. This is essentially as a result of lack of recovery techniques, which is aspired to be considered in the future.

7.2. Datasets and Performance Reports: Scene Character Recognition

First, Quadratic SVM (QSVM), Linear SVM (LSVM) and Cubic SVM (CSVM) are trained on individual feature vectors. Next, the same classifiers are trained on the linear combination of all feature vectors considered as a single feature. Likewise, all the previous SVMs are trained on SIFT to determine its effectiveness based on the classification accuracy that is reported in Table 7.4.

Table 7.4. Classification accuracy on individual feature sets.

	Classification accuracy (%)		
Features	LSVM	QSVM	CSVM
GradientDirection	32.7	51.2	48.1
MaxDistance	37.7	32.6	39.0
DistanceKey	27.9	32.5	31.9
Angle	43.9	45.4	45.1

As given in Table 7.4, Quadratic SVM trained with each individual feature vectors demonstrate the highest classification accuracy. Next, all feature vectors are merged into one global feature of 573 dimension. The same set of SVMs are trained on the final feature vector and the classification accuracy obtained is given in Table 7.5.

Table 7.5. Classification accuracy on GCF and SIFT.

Features	Classification accuracy (%)		
	LSVM	QSVM	CSVM
Proposed(GCF)	63.0	65.3	63.3
SIFT	52	53.7	44

Referring Table 7.5, the following two conclusions can be made about the difference in classification accuracy obtained from SVM of different kernels and the effectiveness of the final feature vector which is a linear combination of the four individual feature vectors.

- QSVM gives the highest classification accuracy when trained on all types of features considered in this thesis.

- The combination of various descriptions resulted in a considerable increase in the classification accuracy.

To strengthen the above conclusions, other classifiers including Complex Trees, Weighted K-Nearest Neighbors and Linear Discriminant Analysis are trained. The results obtained validate the statements about the effectiveness of feature combination and QSVMs.

On the other hand, HOG and proposed feature classification accuracies are presented in table 7.6. Similar to object recognition, HOG features led to better classification accuracy with the largest dimension, lowest prediction speed and more training time.

Table 7.6. Classification accuracy on proposed feature and HOG.

Features	Classification accuracy (%)		
	LSVM	QSVM	CSVM
Proposed(GCF)	63.0	65.3	63.3
HOG	81	82.3	81.7

7.3. ICDAR 2003 Robust Character Recognition

In addition to Char74k dataset, ICDAR2003 dataset is another most widely used standard dataset employed to test an algorithm proposed for segmented character recognition. Therefore, in this thesis too, the same dataset is used and the results are reported in Table 7.7. However, relatively lower classification accuracies are obtained than the results obtained from Chars7k dataset. One of the prominent reasons for the drop in the classification accuracy can be attributed to the size of samples for each character class.

Table 7.7. Classification accuracy on GCF and SIFT and HOG (ICDAR2003).

Features	Accuracy [%]
Proposed(GCF)	56.7
SIFT	49
HOG	75

In addition to the performance of classifiers on features implemented for comparison purpose, comparison with methods such as (23) and (64) is included in table 7.8 .

Table 7.8. Classification accuracy of proposed and unsupervised feature learning

Methods	Accuracy (%)
Proposed(GCF)	65
Neumann and Matas (23)	67
Coates et al.(64)	81

Following the classical quantitative performance measure, other parameters such as feature detection and description time, classifier training and prediction time are also noted to assess and compare the efficiency of GCF and SIFT. The values for these parameters are included in Table 7.9.

Table 7.9. Feature comparison based on computation, training and prediction time

Features	Dimension	Computation time/character image(s)	Training time(s)	Prediction speed(characters/second)
GCF	573	1.68883	66.257	110
SIFT	128	4	68.499	210

Table 7.10. Feature comparison based on computation, training and prediction time

Features	Dimension	Computation time/character image(s)	Training time(s)	Prediction speed(characters/second)
Proposed(GCF)	573	1.68883	66.257	110
HOG	3780	1.356	222.2	17

The proposed global curvature based feature has less feature length than HOG. As is presented in table 7.10, except for the computation time, it is computationally feasible than HOG as well.

CHAPTER 8. CONCLUSION

In this thesis, two methods to detect characters from scene images are presented: methods that are based on clustering and Maximally Unstable Extremal Regions (MUER). In addition to scene character detection methods, a classical, hand-engineered features based technique is also introduced for segmented character recognition. Different from most state-of-the art deep learning based detection and recognition techniques, the methods discussed in this thesis are modeled with bottom-up detection and recognition approaches, each comprising a set of interrelated processes.

During segmentation with K-means clustering, a binary image is generated from scene images having multi-color texts. Individual characters are detected from the corresponding binary image. On the other hand, MUER based detection relies on image region changes where the maximum and minimum pixel values are retrieved as transition points. Like any other bottom-up detection technique, these two methods are also bound to post-processing including filtering and recovery.

Similarly, the MUER based detection stage includes image transformation, transition detection, connected component generation, recovery and filtering. Major contributions include the design of effective connected component generation strategy, which results to most of the candidate character components. In addition, a coarse-to-fine level connected component filtering is also introduced to improve precision without affecting the recall of the algorithm. During transformation, the original scene (multi-color) image is converted into an intermediate representation from which the transition points are retrieved as low and high transition points suggesting locations that the image region is changing from background to foreground and vice-versa. The logarithm and square root functions are used to locate these values from images of various contrasts.

The connected components are initially generated from the combination of the transition points with LOGICAL OR operator. During filtering, specifically at coarse-level, K-means clustering is used as a primary tool to filter non-text lines. Conversely, the bounding box surrounding a candidate character region is employed to return an equivalent region from the combination of high and low transition-based binary images. Connected components are counted in this region and reach at a decision based on the number of components in a particular region.

For the recognition phase, the input is expected to be a segmented character image, from which distinct features are extracted with the application of a novel keypoint selection and feature description strategy. The feature vectors are ultimately used to train SVM of various kernels for character classification. The global feature detection and description technique for the specified task can be considered as a good starting point for further research on global shape descriptors. A segmented character classification accuracy based performance comparison between a very well-known local feature descriptor, SIFT, and the designed feature, GCF, can give an insight into the power of global shape descriptors. In this thesis, In addition to classification accuracy, computation time, training time and prediction time are considered to assess the efficiency of the feature detection and description. Accordingly, the global feature introduced in this thesis, GCF, has a higher efficiency than SIFT. In addition, SIFT takes three times more prediction time than GCF. Also, the total time taken to train SVM using GCF is less than that of SIFT and lastly SIFT requires double prediction time than GCF. In addition, with better feature merging techniques, classification accuracy can be improved. However, current merging technique (simple combination) did not result to any improvement.

Lastly, it is evident from experimental results that both the detection and recognition methods are likely to be associated with some peculiar limitations. First, during image transformation for detection, deciding the contrast of a given input image that is used as a point of reference to select either one of the two transformation functions as high or low is prone to failure in some cases. Second, after detection, there are no precise rules and procedures to filter out false positives as well as recover false negatives.

Rather, some pre-set heuristic rules defined over geometric properties are employed. Since heuristic rules are not flexible and fit to all types of font types or shapes, there are situations where the filtering or recovery worsens precision and recall.

REFERENCES

[1] Wang Y, Shi C, Xiao B, Wang C, Qi C. CRF based text detection for natural scene images using convolutional neural network and context information. Neurocomputing. 2018;295:46–58.

[2] Šarić M. Scene text segmentation using low variation extremal regions and sorting based character grouping. Neurocomputing. 2017;266:56–65.

[3] Lucas SM, Panaretos A, Sosa L, Tang A, Wong S, Young R. ICDAR 2003 robust reading competitions. In Citeseer; 2003. p. 682–7.

[4] Lucas SM. ICDAR 2005 text locating competition results. In IEEE; 2005. p. 80–4.

[5] Shahab A, Shafait F, Dengel A. ICDAR 2011 robust reading competition challenge 2: Reading text in scene images. In IEEE; 2011. p. 1491–6.

[6] Karatzas D, Shafait F, Uchida S, Iwamura M, i Bigorda LG, Mestre SR, et al. ICDAR 2013 robust reading competition. In IEEE; 2013. p. 1484–93.

[7] Karatzas D, Gomez-Bigorda L, Nicolaou A, Ghosh S, Bagdanov A, Iwamura M, et al. ICDAR 2015 competition on robust reading. In IEEE; 2015. p. 1156–60.

[8] Nayef N, Yin F, Bizid I, Choi H, Feng Y, Karatzas D, et al. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt. In IEEE; 2017. p. 1454–9.

[9] Dalal N, Triggs B. Histograms of oriented gradients for human detection. In 2005.

[10] Cortes C, Vapnik V. Support-vector networks. Mach Learn. 1995; 20 (3):273–97.

[11] Zheng Y, Li Q, Liu J, Liu H, Li G, Zhang S. A cascaded method for text detection in natural scene images. Neurocomputing. 2017; 238:307–15.

[12] Pan Y-F, Hou X, Liu C-L. A hybrid approach to detect and localize texts in natural scene images. IEEE Trans Image Process. 2010;20(3)800–13.

[13] Wei Y, Shen W, Zeng D, Ye L, Zhang Z. Multi-oriented text detection from natural scene images based on a CNN and pruning non-adjacent graph edges. Signal Process Image Commun. 2018;64:89–98.

[14] Sun L, Huo Q, Jia W, Chen K. A robust approach for text detection from natural scene images. Pattern Recognit. 2015;48(9):2906–20.

[15] Risnumawan A, Shivakumara P, Chan CS, Tan CL. A robust arbitrary text detection system for natural scene images. Expert Syst Appl. 2014;41[18]:8027–48.

[16] GonzáLez Ál, Bergasa LM. A text reading algorithm for natural images. Image Vis Comput. 2013;31(3):255–74.

[17] Su B, Lu S. Accurate scene text recognition based on recurrent neural network. In Springer; 2014. p. 35–48.

[18] Yang C, Yin X-C, Li Z, Wu J, Guo C, Wang H, et al. AdaDNNs: adaptive ensemble of deep neural networks for scene text recognition. ArXiv Prepr ArXiv171003425. 2017;

[19] Elagouni K, Garcia C, Mamalet F, Sébillot P. Combining multi-scale character recognition and linguistic knowledge for natural scene text OCR. In IEEE; 2012. p. 120–4.

[20] Shi B, Bai X, Yao C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Trans Pattern Anal Mach Intell. 2016;39(11):2298–304.

[21] Wang T, Wu DJ, Coates A, Ng AY. End-to-end text recognition with convolutional neural networks. In IEEE; 2012. p. 3304–8.

[22] Zhan F, Lu S. Esir: End-to-end scene text recognition via iterative image rectification. In 2019. p. 2059–68.

[23] Neumann L, Matas J. A method for text localization and recognition in real-world images. In Springer; 2010. p. 770–83.

[24] Zhang H, Zhao K, Song Y-Z, Guo J. Text extraction from natural scene image: A survey. Neurocomputing. 2013;122:310–23.

[25] Zhu Y, Yao C, Bai X. Scene text detection and recognition: Recent advances and future trends. Front Comput Sci. 2016;10(1):19–36.

[26] Liu X, Meng G, Pan C. Scene text detection and recognition with advances in deep learning: a survey. Int J Doc Anal Recognit IJDAR. 2019;22(2):143–62.

[27] Patterson J, Gibson A. Deep learning: A practitioner's approach. O'Reilly Media, Inc.; 2017.

[28] Heaton J. Artificial intelligence for humans. Heaton Research, Incorporated; 2013.

[29] Matas J, Chum O, Urban M, Pajdla T. Robust wide-baseline stereo from maximally stable extremal regions. Image Vis Comput. 2004;22(10):761–7.

[30] Neumann L, Matas J. Real-time scene text localization and recognition. In IEEE; 2012. p. 3538–45.

[31] Neumann L, Matas J. Efficient scene text localization and recognition with local character refinement. In IEEE; 2015. p. 746–50.

[32] Baran R, Partila P, Wilk R. Automated text detection and character recognition in natural scenes based on local image features and contour processing techniques. In Springer; 2018. p. 42–8.

[33] Huang X, Shen T, Wang R, Gao C. Text detection and recognition in natural scene images. In IEEE; 2015. p. 44–9.

[34] Yin X-C, Pei W-Y, Zhang J, Hao H-W. Multi-orientation scene text detection with adaptive clustering. IEEE Trans Pattern Anal Mach Intell. 2015;37(9):1930–7.

[35] Gomez L, Karatzas D. Object proposals for text extraction in the wild. In IEEE; 2015. p. 206–10.

[36] Chen H, Tsai SS, Schroth G, Chen DM, Grzeszczuk R, Girod B. Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In IEEE; 2011. p. 2609–12.

[37] Yin X-C, Yin X, Huang K, Hao H-W. Robust text detection in natural scene images. IEEE Trans Pattern Anal Mach Intell. 2013;36(5):970–83.

[38] Shi C, Wang C, Xiao B, Zhang Y, Gao S. Scene text detection using graph model built upon maximally stable extremal regions. Pattern Recognit Lett. 2013;34(2):107–16.

[39] Koo HI, Kim DH. Scene text detection via connected component clustering and nontext filtering. IEEE Trans Image Process. 2013;22(6):2296–305.

[40] Ye Q, Doermann D. Scene text detection via integrated discrimination of component appearance and consensus. In Springer; 2013. p. 47–59.

[41] Neumann L, Matas J. Text localization in real-world images using efficiently pruned exhaustive search. In IEEE; 2011. p. 687–91.

[42] He T, Huang W, Qiao Y, Yao J. Text-attentional convolutional neural network for scene text detection. IEEE Trans Image Process. 2016;25(6):2529–41.

[43] Milyaev S, Barinova O, Novikova T, Kohli P, Lempitsky V. Image binarization for end-to-end text understanding in natural images. In IEEE; 2013. p. 128–32.

[44] Belhedi A, Marcotegui B. Adaptive scene-text binarisation on images captured by smartphones. IET Image Process. 2016;10(7):515–23.

[45] Wu H, Zou B, Zhao Y, Chen Z, Zhu C, Guo J. Natural scene text detection by multi-scale adaptive color clustering and non-text filtering. Neurocomputing. 2016;214:1011–25.

[46] Wang X, Song Y, Zhang Y. Natural scene text detection with multi-channel connected component segmentation. In IEEE; 2013. p. 1375–9.

[47] Pise A, Ruikar S. Text detection and recognition in natural scene images. In IEEE; 2014. p. 1068–72.

[48] Wang R, Sang N, Gao C. Text detection approach based on confidence map and context information. Neurocomputing. 2015;157:153–65.

[49] Fabrizio J, Marcotegui B, Cord M. Text detection in street level images. Pattern Anal Appl. 2013;16(4):519–33.

[50] Yi C, Tian Y. Text string detection from natural scenes by structure-based partition and grouping. IEEE Trans Image Process. 2011;20(9):2594–605.

[51] Epshtein B, Ofek E, Wexler Y. Detecting text in natural scenes with stroke width transform. In IEEE; 2010. p. 2963–70.

[52] Yao C, Bai X, Liu W, Ma Y, Tu Z. Detecting texts of arbitrary orientations in natural images. In IEEE; 2012. p. 1083–90.

[53] Yao C, Bai X, Liu W. A unified framework for multioriented text detection and recognition. IEEE Trans Image Process. 2014;23(11):4737–49.

[54] Dey S, Shivakumara P, Raghunandan K, Pal U, Lu T, Kumar GH, et al. Script independent approach for multi-oriented text detection in scene image. Neurocomputing. 2017;242:96–112.

[55] Huang W, Lin Z, Yang J, Wang J. Text localization in natural images using stroke feature transform and text covariance descriptors. In 2013. p. 1241–8.

[56] He T, Huang W, Qiao Y, Yao J. Accurate text localization in natural image with cascaded convolutional text network. ArXiv Prepr ArXiv160309423. 2016;

[57] Jaderberg M, Simonyan K, Vedaldi A, Zisserman A. Reading text in the wild with convolutional neural networks. Int J Comput Vis. 2016;116(1):1–20.

[58] Neumann L, Matas J. Scene text localization and recognition with oriented stroke detection. In 2013. p. 97–104.

[59] Niblack W. An introduction to digital image processing. Strandberg Publishing Company; 1985.

[60] Zhao Z, Fang C, Lin Z, Wu Y. A robust hybrid method for text detection in natural scenes by learning-based partial differential equations. Neurocomputing. 2015;168:23–34.

[61] Huang W, Qiao Y, Tang X. Robust scene text detection with convolution neural network induced mser trees. In Springer; 2014. p. 497–511.

[62] Bissacco A, Cummins M, Netzer Y, Neven H. Photoocr: Reading text in uncontrolled conditions. In 2013. p. 785–92.

[63] Zhang Z, Shen W, Yao C, Bai X. Symmetry-based text line detection in natural scenes. In 2015. p. 2558–67.

[64] Coates A, Carpenter B, Case C, Satheesh S, Suresh B, Wang T, et al. Text Detection and Character Recognition in Scene Images with Unsupervised Feature Learning. In 2011. p. 440–5.

[65] Ma J, Shao W, Ye H, Wang L, Wang H, Zheng Y, et al. Arbitrary-oriented scene text detection via rotation proposals. IEEE Trans Multimed. 2018;20(11):3111–22.

[66] Qin S, Manduchi R. Cascaded segmentation-detection networks for word-level text spotting. In IEEE; 2017. p. 1275–82.

[67] Qin H, Zhang H, Wang H, Yan Y, Zhang M, Zhao W. An Algorithm for Scene Text Detection Using Multibox and Semantic Segmentation. Appl Sci. 2019;9(6):1054.

[68] He W, Zhang X-Y, Yin F, Liu C-L. Deep direct regression for multi-oriented scene text detection. In 2017. p. 745–53.

[69] Jaderberg M, Vedaldi A, Zisserman A. Deep features for text spotting. In Springer; 2014. p. 512–28.

[70] Liu Y, Jin L. Deep matching prior network: Toward tighter multi-oriented text detection. In 2017. p. 1962–9.

[71] Busta M, Neumann L, Matas J. Deep textspotter: An end-to-end trainable scene text localization and recognition framework. In 2017. p. 2204–12.

[72] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. In 2016. p. 779–88.

[73] Gupta A, Vedaldi A, Zisserman A. Synthetic data for text localisation in natural images. In 2016. p. 2315–24.

[74] Zhong Z, Jin L, Zhang S, Feng Z. Deeptext: A unified framework for text proposal generation and text detection in natural images. ArXiv Prepr ArXiv160507314. 2016;

[75] Jiang Y, Zhu X, Wang X, Yang S, Li W, Wang H, et al. R2cnn: Rotational region cnn for orientation robust scene text detection. ArXiv Prepr ArXiv170609579. 2017;

[76] Zhou X, Yao C, Wen H, Wang Y, Zhou S, He W, et al. EAST: an efficient and accurate scene text detector. In 2017. p. 5551–60.

[77] Liu X, Liang D, Yan S, Chen D, Qiao Y, Yan J. Fots: Fast oriented text spotting with a unified network. In 2018. p. 5676–85.

[78] Dai Y, Huang Z, Gao Y, Xu Y, Chen K, Guo J, et al. Fused text segmentation networks for multi-oriented scene text detection. In IEEE; 2018. p. 3604–9.

[79] Yang Q, Cheng M, Zhou W, Chen Y, Qiu M, Lin W, et al. Inceptext: A new inception-text module with deformable psroi pooling for multi-oriented scene text detection. ArXiv Prepr ArXiv180501167. 2018;

[80] Liao M, Shi B, Bai X, Wang X, Liu W. Textboxes: A fast text detector with a single deep neural network. In 2017.

[81] Bazazian D, Gomez R, Nicolaou A, Gomez L, Karatzas D, Bagdanov AD. Improving text proposals for scene images with fully convolutional networks. ArXiv Prepr ArXiv170205089. 2017;

[82] Zhang Z, Zhang C, Shen W, Yao C, Liu W, Bai X. Multi-oriented text detection with fully convolutional networks. In 2016. p. 4159–67.

[83] Deng D, Liu H, Li X, Cai D. Pixellink: Detecting scene text via instance segmentation. In 2018.

[84] He P, Huang W, He T, Zhu Q, Qiao Y, Li X. Single shot text detector with regional attention. In 2017. p. 3047–55.

[85] Wu Y, Natarajan P. Self-organized text detection with minimal post-processing via border learning. In 2017. p. 5000–9.

[86] Li H, Wang P, Shen C. Towards end-to-end text spotting with convolutional recurrent neural networks. In 2017. p. 5238–46.

[87] De Campos TE, Babu BR, Varma M. Character recognition in natural images. VISAPP 2. 2009;7.

[88] Sinha Y, Jain P, Kasliwal N. Comparative study of preprocessing and classification methods in character recognition of natural scene images. In: Machine Intelligence and Signal Processing. Springer; 2016. p. 119–29.

[89] Su B, Lu S, Tian S, Lim JH, Tan CL. Character recognition in natural scenes using convolutional co-occurrence hog. In IEEE; 2014. p. 2926–31.

[90] Shi C, Wang Y, Jia F, He K, Wang C, Xiao B. Fisher vector for scene character recognition: A comprehensive evaluation. Pattern Recognit. 2017;72:1–14.

[91] Wang Y, Shi C, Wang C, Xiao B, Qi C. Multi-order co-occurrence activations encoded with Fisher Vector for scene character recognition. Pattern Recognit Lett. 2017;97:69–76.

[92] Roy S, Roy PP, Shivakumara P, Louloudis G, Tan CL, Pal U. HMM-based multi oriented text recognition in natural scene image. In IEEE; 2013. p. 288–92.

[93] Yang C-S, Yang Y-H. Improved local binary pattern for real scene optical character recognition. Pattern Recognit Lett. 2017;100:14–21.

[94] Higa K, Hotta S. Local Subspace Classifier with Transformation Invariance for Appearance-Based Character Recognition in Natural Images. In IEEE; 2013. p. 533–7.

[95] Liu X, Lu T. Natural scene character recognition using markov random field. In IEEE; 2015. p. 396–400.

[96] Wan Y, Xie F, Liu Y, Bai X, Yao C. 2D-CTC for Scene Text Recognition. 2019.

[97] Gao Y, Huang Z, Dai Y. Double Supervised Network with Attention Mechanism for Scene Text Recognition. ArXiv Prepr ArXiv180800677. 2018;

[98] Wang Q, Jia W, He X, Lu Y, Blumenstein M, Huang Y. FACLSTM: ConvLSTM with Focused Attention for Scene Text Recognition. ArXiv Prepr ArXiv190409405. 2019;

[99] Cheng Z, Bai F, Xu Y, Zheng G, Pu S, Zhou S. Focusing attention: Towards accurate text recognition in natural images. In 2017. p. 5076–84.

[100] He P, Huang W, Qiao Y, Loy CC, Tang X. Reading scene text in deep convolutional sequences. In 2016.

[101] Wang K, Babenko B, Belongie S. End-to-end scene text recognition. In IEEE; 2011. p. 1457–64.

[102] Mishra A, Alahari K, Jawahar C. Scene text recognition using higher order language priors. In 2012.

[103] Jaderberg M, Simonyan K, Vedaldi A, Zisserman A. Synthetic data and artificial neural networks for natural scene text recognition. ArXiv Prepr ArXiv14062227. 2014;

[104] Godil A, Bostelman R, Shackleford W, Hong T, Shneier M. Performance metrics for evaluating object and human detection and tracking systems. 2014.

[105] Wolf C, Jolion J-M. Object count/area graphs for the evaluation of object detection and segmentation algorithms. Int J Doc Anal Recognit IJDAR. 2006;8(4):280–96.

[106] Freeman H, Shapira R. Determining the minimum-area encasing rectangle for an arbitrary closed curve. Commun ACM. 1975;18(7):409–13.

[107] Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A. The pascal visual object classes (voc) challenge. Int J Comput Vis. 2010;88(2):303–38.

[108] Liu Y, Jin L, Xie Z, Luo C, Zhang S, Xie L. Tightness-aware Evaluation Protocol for Scene Text Detection. In 2019. p. 9612–20.

[109] Lee CY, Baek Y, Lee H. TedEval: A Fair Evaluation Metric for Scene Text Detectors. ArXiv Prepr ArXiv190701227. 2019;

[110] Lin H, Yang P, Zhang F. Review of Scene Text Detection and Recognition. Arch Comput Methods Eng. 2019;1–22.

[111] Sonka M, Hlavac V, Boyle R. Image processing, analysis, and machine vision. Cengage Learning; 2014.

[112] Kumar RM, Sreekumar K. A survey on image feature descriptors. Int J Comput Sci Inf Technol. 2014;5:7668–73.

[113] Tuytelaars T, Mikolajczyk K. Local invariant feature detectors: a survey. Found Trends® Comput Graph Vis. 2008;3(3):177–280.

[114] Salahat E, Qasaimeh M. Recent advances in features extraction and description algorithms: A comprehensive survey. In IEEE; 2017. p. 1059–63.

[115] Thomas A, Sreekumar K. A survey on image feature descriptors-color, shape and texture. Int J Comput Sci Inf Technol. 2014;5(6):7847–50.

[116] Yang M, Kpalma K, Ronsin J. A survey of shape feature extraction techniques. 2008;

[117] Sebastian V, Unnikrishnan A, Balakrishnan K. Gray level co-occurrence matrices: generalisation and some new features. ArXiv Prepr ArXiv12054831. 2012;

[118] Leutenegger S, Chli M, Siegwart R. BRISK: Binary robust invariant scalable keypoints. In Ieee; 2011. p. 2548–55.

[119] Alahi A, Ortiz R, Vandergheynst P. Freak: Fast retina keypoint. In Ieee; 2012. p. 510–7.

[120] Rublee E, Rabaud V, Konolige K, Bradski GR. ORB: An efficient alternative to SIFT or SURF. In Citeseer; 2011. p. 2.

[121] Bay H, Tuytelaars T, Van Gool L. Surf: Speeded up robust features. In Springer; 2006. p. 404–17.

[122] Lowe DG. Distinctive image features from scale-invariant keypoints. Int J Comput Vis. 2004;60(2):91–110.

[123] Assirati L, Silva NR da, Berton L, Lopes A de A, Bruno OM. Performing edge detection by difference of gaussians using q-gaussian kernels. In IOP Publishing; 2014. p. 012020.

[124] Mohammed M, Khan MB, Bashier EBM. Machine learning: algorithms and applications. Crc Press; 2016.

[125] Heaven D. Why deep-learning AIs are so easy to fool. 2019;

[126] LeCun Y, Bengio Y, Hinton G. Deep learning. nature. 2015;521(7553):436.

[127] Wetherell C. The Log Percent (L%): An Absolute Measure of Relative Change. Hist Methods. 1986;19(1):25.

[128] Mohideen F, Rodrigo R. Curvature Based Robust Descriptors. In 2012. p. 1–11.

[129] Dalle D. Comparison of numerical techniques for Euclidean curvature. Rose-Hulman Undergrad Math J. 2006;7(1):12.

[130] Auria L, Moro RA. Support vector machines (SVM) as a technique for solvency analysis. 2008;

RESUME

Belaynesh Chekol is born in Ethiopia. She attended her primary, secondary and high school in Motta. In 2006, she graduated from Adama University with a bachelor of education degree in information technology and computer science. For the next three years, she had worked as a graduate assistant in Ambo University, Ethiopia, in the department of computer science. From 2009-2011, she attended her education in Osmania University, India, from which she graduated with a master degree in computer science. Following her graduation, she has worked as an assistant lecturer in Ambo University for the following two years. Starting from 2014 she is pursuing her PhD in Sakarya University, Turkey, in the department of Computer Engineering.