Multi-lingual scene text detection based on convolutional neural networks

Wafa Khlif

UNIVERSITY OF LA ROCHELLE

DOCTORAL SCHOOL EUCLIDE

L3i LABORATORY

THESIS by :

Wafa KHLIF

defended on : **Thursday, 2 June 2022**
carried out at : **Laboratoire Informatique, Image et Interaction (L3i)**
&
Research Groups in Intelligent Machines (REGIM)

for the award of the doctoral degree : **DOCTOR IN COMPUTER SCIENCES**
Discipline : **Computer Science and Applications**

Multi-lingual Scene Text Detection Based on Convolutional Neural Networks

---

Reviewers	Prof. Dr. Jean-Yves Ramel	Laboratory LIFAT, University of Tours
	Prof. Dr. Najoua Ben Amara	Laboratory LATIS, University of Sousse
Examiners	Prof. Dr. Yacine Ghamri-Doudane	Laboratory L3i, University of La Rochelle
	Prof. Dr. Véronique Eglin	Laboratory LIRIS, University of Lyon
	Dr. Nibal Nayef	Researcher at MyScript Company
Invited	Dr. Ikram Moalla	Al Baha University and University of Sfax
Directors	Prof. Dr. Jean-Christophe Burie	Laboratory L3i, University of La Rochelle
	Prof. Dr. Adel Alimi	Laboratory ReGim, University of Sfax

Résumé

Les smartphones sont devenus des dispositifs essentiels pour capturer les images rapidement et facilement. Leur usage par le grand public a conduit à une croissance énorme de la quantité de données multimédia disponible sur le web. Les publicités, les photos de vacances, les cartes de visite et les articles de journaux sont de nos jours couramment numérisés par ces appareils. Récemment, les grandes entreprises et même les gouvernements ont trouvé de nouvelles façons d'exploiter les données des utilisateurs qu'ils ont recueillies dans un but commercial mais aussi pour des question de sécurité. La plupart de ces développements ont permis aux personnes et aux organisations de faire face à d'énormes quantités de données multimédia présentes sur les réseaux, et notamment les informations de type texte, image, vidéo et audio. Les images et les données vidéo ne sont pas structurées et sont particulièrement difficiles à analyser et à comprendre par les ordinateurs. Bien que des outils de recherche soient disponibles sur le réseau, la compréhension et l'exploration approfondies du contenu de ces images est encore difficile. Avec de telles collections de données, le domaine de la récupération d'informations (IR) a évolué rapidement pour fournir des services numériques enrichis et personnalisés pour retrouver des informations pertinentes.

Cette thèse propose des approches de détection de texte par des techniques d'apprentissage profond pour explorer et récupérer des contenus faiblement structurés dans des images de scène naturelles. Les contenus faiblement structurés - ou non structurés - concernent spécifiquement un grand nombre d'images que l'on peut désormais trouver sur les réseaux sociaux, qui ont été pour la plupart capturées par des appareils mobiles ou synthétisées par des outils de retouche d'images. L'analyse du contenu de ces images contribuera au développement de la prochaine génération de moteurs de recherche. L'atteinte de cet objectif sera très utile pour des applications comme la cybersécurité et l'exploitation des données dans un but commercial, ainsi que pour des applications sociales comme l'orientation interactive des touristes. Un aperçu des principales contributions de cette thèse est donné ci-après :

Ces travaux de recherche proposent, dans un premier temps, une méthode de détection de texte dans des images de scène naturelle basée sur une analyse multi-niveaux des composantes connexes (CC) et l'apprentissage des caractéristiques du texte par un réseau de neurones convolutionnel (CNN), suivie d'un regroupement des zones de texte détectées par une méthode à base de graphes. La méthode effectue une extraction redondantes des textes et des composantes non textuelles par une binarisation multi-niveaux afin de minimiser la perte de tout candidat de texte potentiel. Les caractéristiques des composantes texte brut/non-texte obtenues à différents niveaux de granularité sont apprises via un CNN. Ces deux modules évitent l'utilisation de prétraitements complexes pour trouver les candidats initiaux, ainsi que des approches artisanales pour classer ces candidats en tant que texte ou non-texte. Les éléments classés en tant que texte à différents niveaux de granularité sont regroupés dans un graphe en fonction du chevauchement de

leurs boites englobantes. Cette stratégie élimine les éléments de texte redondants et crée des mots ou des lignes de texte. Lors des expérimentation sur des bases de données publiques telles que la base de données "Robust Reading Competition" pour les images de scènes naturelles, la méthode proposé obtient de meilleurs résultats de détection par rapport aux méthodes de la littérature utilisant des approches similaires. En plus de son efficacité, la méthode peut être facilement adaptée pour détecter le texte multilingue ou multi-orienté car elle fonctionne avec des éléments de bas niveau, et elle ne nécessite pas que ces éléments soient des caractères. A notre connaissance, il s'agit d'une approche originale par rapport aux méthodes de la littérature.

Afin d'améliorer la performance de la méthode de détection dans le traitement de textes multilingues, une autre méthode est présentée dans cette thèse inspirée du système YOLO : Système de détection d'objets en temps réel. Le systéme réalise la detection du texte et l'identification du script simultanément. Nous avons fait deux expérimentations. La première expérimentation consiste à considérer la tâche de détection de texte multiscript comme un problème de détection d'objets, où l'objet est le script du texte. La détection de texte et l'identification de script Latin/Arabe sont réalisées avec une approche holistique en utilisant un réseau neuronal convolutionnel unique où la donnée d'entrée est l'image complète et les sorties sont les zones de texte et le script du texte détecté. Les évaluations expérimentales de la méthode sur trois jeux de données (ICDAR15 pour les images de scènes focalisées FSN, images nativement numériques BDI et ARASTI), présentent une précision de classification de 92,47% et un taux d'erreur d'identification de script inférieur à 7%. La deuxième expérimentation est une extension de la première méthode appliqué sur 9 scripts. Les caractéristiques et leurs scripts sont appris via un réseau neuronal convolutif unique et le CNN constitue le composant principal dans les deux méthodes.

Les évaluations expérimentales de ces approches sont réalisées sur le jeu de données MLT (Multi-Lingual Text dataset). Dans le cadre du projet ANR AUDINM (Analysis and Understanding of Document Images in Network Media), nous avons contribué à la création de ce nouveau jeu de données. Il est composé d'images de scènes naturelles contenant du texte, tels que des panneaux de circulation, des panneaux publicitaires, des noms de magasins, des véhicules, d'images extraites des réseaux sociaux. Ce type d'images représente l'un des types d'images les plus fréquemment rencontrés sur Internet, à savoir les images avec du texte incorporé dans les réseaux sociaux.

Les évaluations montrent que nos approches sont capables de détecter correctement le texte multilingue présent dans l'image. Ces approches ont été testées sur plusieurs jeux de données ayant des caractéristiques différentes. Par rapport aux méthodes de la littérature, nos approches offrent de meilleures performances en termes de robustesse lorsqu'elles sont appliquées sur différents types d'images.

Abstract

Smartphones have become essential devices for capturing easily and quickly images. This has led to a huge growth in the amount of multimedia data on the web. Advertisements, holiday pictures, business cards and newspaper articles are nowadays commonly digitized by these devices. Recently, large-scale commercial search companies and governments have found new ways to exploit the collected user data for commercial aspects and security/safety considerations. Most of these developments have allowed people and organizations to face huge amounts of multimedia data in networks, including text, image, video and audio. Image and video data are unstructured and are particularly hard to analyze and understand by computers. Though techniques for network search and information retrieval are available, the deep understanding and mining of the contents of such images are still lacking. With such large data collections, the field of Information Retrieval (IR) has evolved rapidly to provide enriched and personalized digital services for retrieving information for different purposes.

This dissertation explores text detection approaches via deep learning techniques towards achieving the goal of mining and retrieval of weakly structured contents in scene images. Weakly structured – or non-structured – contents concern specifically a large set of images that can now be found on social networks, which have mostly been captured by mobile devices or synthesized by image-editing tools. Analyzing the contents of those images will help in the development of the next generation of search engines. Achieving this goal will be very useful for applications like cyber security, commercial data mining and social applications such as interactive tourists' guidance. An overview of the key contributions of this dissertation is given below.

First, this dissertation presents a method for detecting text in scene images based on multi-level connected component (CC) analysis and learning text component features via convolutional neural networks (CNN), followed by a graph-based grouping of overlapping text boxes. The method performs the extraction of redundant text and non-text components at multiple binarization levels to minimize the loss of any potential text candidates. The features of the resulting raw text/non-text components of different granularity levels are learned via a CNN. Those two modules eliminate complex ad-hoc preprocessing steps for finding initial candidates, and the need of hand-designed features to classify such candidates into text or non-text. The components classified as text at different granularity levels are grouped in a graph based on the overlap of their extended bounding boxes, then, the connected graph components are retained. This eliminates redundant text components and form words or text lines. When evaluated on standard databases such as the "Robust Reading Competition" dataset for natural scene images, the method achieved better detection results compared to the state-of-the-art methods using similar techniques. In addition to its efficacy, the method can be easily adapted to detect multi-oriented or multi-lingual text as it operates at low level on the initial components, and it does not require such components to be characters.

In order to improve the performance of the detection method when dealing with multi-lingual text, two novel methods are presented in this research work. Both methods are inspired from YOLO: Real-Time Object Detection system. Both methods perform text detection and script identification simultaneously. The first method is a joint text detection and script identification approach by casting the multi-script text detection task as an object detection problem, where the object is the script of the text. Joint text detection and Latin/Arabic script identification strategy is realized in a holistic approach using a single convolutional neural network where the input data is the full image and the outputs are the text bounding boxes and the script of the detected text. The experimental evaluation of the method on three datasets (ICDAR15 for Focused Scene Text FST, born-digital images BDI and ARASTI dataset), presents classification accuracy of 92.47% and a script identification error rate lower than 7%. The second method is an extension of the first work. Textual feature extraction and script classification are performed jointly in one step. The relative features and their scripts are learned via a single convolutional neural network and this CNN presents the main component in both methods.

The experimental evaluation of these methods is also performed on the Multi-Lingual Text MLT dataset. As part of the ANR AUDINM project (Analysis and Understanding of Document Images in Network Media), we contributed in building a new dataset. It is constituted of natural scene images with embedded text, such as street signs, street advertisement boards, shops names, passing vehicles, user photos in microblog. This kind of images represents one of the mostly encountered image types on the internet which are the images with embedded text in social media.

Overall, these contributions could be considered as significant steps in the right direction of the research of information retrieval from scene images with multi-lingual text. The assessments show that our approaches are able to properly detect the embedded multi-lingual text in the image. These approaches are tested in different datasets with different characteristics. In comparison with state-of-the-art methods, our approaches give competitive performance in terms of robustness with applications to various types of images.

Acknowledgments

First of all I would like to thank the members of the jury Professors Jean-Yves Ramel, Najoua Ben Amara, Veronique Eglin, Yacine Ghamri-Doudane and Nibal Nayef for having accepted to assess my thesis.

Writing my thesis has been a very challenging experience and a long chapter in my career, it has been possible thanks to the support and guidance that I received from many precious people.

I would like to express my deepest gratitude to the people who believed on me and helped me to achieve this work. First I would like to express my heartfelt gratitude to my supervisors, Professor Jean-Christophe Burie and Professor Adel Alimi for the continuous academic and emotional support. This thesis was made possible due to their constant encouragement and belief in me. My sincere gratitude also goes to my advisors Doctor Nibal Nayef and Doctor Ikram Moalla. My discussions with them have been interesting and inspiring. And their guidance helped me in all the time of research.

A special acknowledgment goes to my laboratory the L3i, for giving me the comfort and material support necessary for a thesis in computer science. I also thank REGIM laboratory for accepting my candidature to do this thesis. Both of them have given me high confidence, which is definitely the greatest reward of my endeavors. It is my honor and pleasure to work with great teams and that is definitely priceless. I would also thank IUT for allowing me to teach within their walls and to exercise this beautiful facet of the teaching-research profession. A special thanks to the CASIA (Institute of Automation of the Chinese Academy of Sciences) to welcome me for four months in the NLPR (National Laboratory of Pattern Recognition). Thanks to Professor Cheng-Lin Liu and his team for the organization of that stay and the very warm welcome they gave us.

I would like to thank my good friends in L3i Lab for being helpful and nice, especially during the times when I was stressed about my thesis progress. Thanks for contributing to my work through many comments on the thesis, and thanks for all the good times we spent together. A lot of thanks and gratitude to my 127 open space friends.

Most of all, I would like to thank my parents, for always believing in me and encouraging me to achieve my dreams. My sister and my brother, who have each time been a very precious source of support and for their legendary patience. Special thanks go to my husband for his tremendous help and support, both emotional as well as intellectual. To my lovely daughter, you are my source of motivation. Thank you grand-mom, wish you were here with us, your prayers for me were what sustained me thus far.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Context and Positioning

Thanks to the recent breakthroughs of the new digital economy, many developments have been made to provide enhanced and customized digital services for information retrieval (IR) on the Internet. These advanced digital services provide useful information and enrich the lives of Internet users. The field of IR has been continuously reshaping and evolving; with larger data collections, more powerful computers, broadband and mobile internet are widely available, complex interactive searches can now be performed on PCs or mobile devices, etc.

Recently, the major market research firms and governments have found new ways to use the data they collect from users for commercial and safety reasons. Most of these developments have had the effect of exposing individuals and organizations to large amounts of multimedia data on networks, including text, images, videos and audios. Image and video data are unstructured and are particularly difficult for computers to analyze and understand. Although there are techniques to search and retrieve network information, there is still a lack of understanding and overall use of the content of these images.

Document images are an important category of networked media data. The volume of this type of data is rapidly increasing due to the growing use of smartphones and social media sites (Facebook, Twitter, LinkedIn, Weibo, etc.). A study [4] shows that 17% of web pages contain text in images, while 76% of text in images occurs only in images. This shows the need to recognize text in images. A keynote speech delivered by the chairman of Baidu Co. Ltd, showed that text-in-image recognition (TIR) is one of the nine challenging problems for web search [5]. Some software tools have been developed to extract text from web images [6], but they are not very powerful for cluttered images.

Images that contain textual information are commonly referred to document images. Potential applications of web document analysis include business data mining (processing a large set of advertising images), information extraction (extracting different types of textual information from images) and Cyber security (searching for sensitive content in images). Tasks like text detection and recognition have gained much popularity in the domain of text analysis systems as they pave the way for a number of real-time-based applications such as assistive methods for tourists or mobile transliteration technologies, etc.

Document images can be divided into four main categories: Scene text images, scanned paper documents, camera-captured document images and synthesized (initially digital) documents. As both scene text images and synthesized documents (originally digital) are increasingly popular in networked media and present new technical challenges compared to traditional paper documents, in this thesis we primarily consider these two categories of documents :

- Scene text images can be captured by mobile cameras and streamed over the network (Figure 1.1 shows some images with embedded text).
- Born digital documents are digital objects created on computers using writing and editing tools. Most of them provide only pixel-level data, such as raster images. There are virtually an unlimited number of images of digital documents that we can find in commercial advertisements (Figure 1.2) and in personal communications on social media (e.g. the Weibo images in Figure 1.3).

Figure 1.1: Scene text images with embedded text

Though the analysis of scene text images and born-digital documents are needed in data mining and retrieval process in network media. Many research efforts have been devoted to this field. The available technologies cannot yet satisfy applications' needs. There are a lot of technical challenges in the related problems (will be detailed in the next section), including scene text and graphics recognition, layout analysis of weakly structured documents or new tasks as multilingual text detection and recognition.

In order to contribute in solving some challenging problems in the field of information retrieval, many projects have been proposed. This dissertation presents a group of methods that contributes to providing such needed solutions. This research work is part

Figure 1.2: Advertisement images

Figure 1.3: Images from Weibo (micro-blog)

of the ANR AUDINM project : Analysis and Understanding of Document Images in Network Media. The objective of this project is to develop more accurate, efficient and robust solutions for the mining and the retrieval of heterogeneous documents in network media. It mainly focuses on weakly structured documents such as born-digital and scene images with embedded text. The project is organized into five scientific work packages:

- Fast image categorization
- Scene Text Detection and Extraction

- Multi-lingual text recognition
- Layout Analysis and Graphics Recognition
- Contextual interpretation and information integration

More details can be found in the website of the project `http://audinm.univ-lr.fr/`

1.2 Challenges of Text Detection

Scene text images and born-digital document images are popular in network media and bring new challenges compared to scanned paper documents. In this work we focus on these two categories. The main difficulties of scene text detection include cluttered background, perspective distortion, lighting variation, defocus and variations of text font type, size and color. Figure 1.4 shows some examples of challenges. Due to this natural presence, such text can manifest itself in a wide range of conditions, depending upon several factors related to the scene and the acquisition process. Compared to paper documents, the segmentation of texts from the scene image is challenging and remains unsolved. The detection and localization of scene texts, due to the image defocus and lighting variation, is more difficult than text detection in paper documents. The main difficulties of born-digital documents include mixed graphics and texts, complex layouts, variations of color and font types and sometimes very low resolution. Particularly, the complex structures of graphical elements and layouts make the structural interpretation and semantic understanding of the whole document very hard. Either scene text images or born-digital documents can be generated and distributed via the Web over the world.

Besides, the multilingual nature of texts (language unknown a priori) poses another challenge. In the absence of a specific context, the text (words or lines) to be recognized in the same image may belong to different scripts. A script refers to a collection of characters used to write one or more languages. For example : Latin script languages presents 52 letters including both lower and upper case in addition to 10 numerals and punctuation marks. However, other scripts such as Arabic, Chinese or Indian have a huge number of character classes and much more of inter-class similarities. These facts present extra challenges for text detection or script identification task.

Furthermore, deep learning works best when it has lots of quality data available for the training of the network. The performance grows as the data available is better labeled

Figure 1.4: Examples of challenges faced when dealing with scene text images.

and significant. When enough quality data is not simply fed to the network, learned features will not be significant and the learning process can fail quite badly. However, it is not enough to have lots of data, it is better to find the best way to present the data to the deep learning system. A study of the existing datasets used in the field of scene text images analysis is presented in Section 2.5.

1.3 Contributions of this Thesis

Our goal in this work is to develop a multi-lingual text detection system. We address the particular tasks of joint text detection and script identification in scene and born digital images. The process should be done while being robust to all the challenges related to the text and the acquisition conditions. We consider both component-based and region-

based approaches, combining the multiple features of color, edges, texture as well as learned features to estimate the local text confidence. Based on the confidence map, candidate text components are segmented and verified according to text appearance. This dissertation presents methods that contribute to providing such needed solutions. Our goal was to achieve better results than state-of-the-art methods using deep learning techniques and to better use the available data. The contributions of our research works are :

- The creation of a novel "Multi-Lingual Text" dataset

  This dataset will help to systematically benchmark the methods and push the state-of-the-art forward. Building this dataset has required an important team work. The members participated in the collection and labeling of the data, the setting of the rules for the annotation and the preparation of the different competitions that we have organized. We developed some algorithms for filtering and preparing the images to be published. We set a bunch of rules for the annotation process specially for the Arabic script (due to the numerous exceptions we found in the captured images). Building this dataset was very important in order to help the research community in standardizing the evaluation of multi-lingual text detection and recognition systems.

- The development of a component-based method for text detection.

  The main idea is to detect text in natural scene and born-digital images using convolutional neural networks for learning the features of text components. The developed method achieved better detection accuracy on standard databases of images from different types. We introduce a multi-level connected component extraction step. An adaptive binarization for dealing with complex backgrounds and color variations is applied on the image to improve the quality of the extracted connected components. We developed a binarization method for correcting the text strokes and get the complete contours of the different characters. Then, we learn the text features from the raw components. The architecture of the used convolutional neural network was designed from scratch. Finally, we developed a graph-based grouping method for dealing with the multi-level text connected components.

- The development of a region-based method for a joint text detection and script identification approach.

  The method was built using a convolutional neural network inspired from an object detection system. The method achieved significantly lower error rates on different standard databases for both tasks. Following an integrated methodology we proceed the text detection task by casting it to an object detection problem. In this work we tried to take advantage of the advancement in the field of object detection to present a robust system. The learning is carried out in several steps to better learn the features of each text script. The script identification problem is reformulated as an object detection problem. This new representation of the data generates different types of features and allows to incorporate higher level information on the features such as weights and adjacency information. Moreover, the script identification task is performed on the entire image, without the need to crop the image and to prepare the cropped words.

The details of each contribution will be discussed in the rest of the chapters of this dissertation as well as the experiments and evaluations necessary to prove their relevance.

1.4 Thesis Organization

In Chapter 2, we present a review of the existing techniques for text detection in the wild. The existing methods could be classified into two categories: methods before the area of deep learning and approaches based on deep learning. Then we give a summary of the popular methods used for the script identification task. Finally, we present a review of the existing datasets used for the text detection task and explain how the performance evaluation is done.

We detail in Chapter 3 the Multi-lingual text database. This dataset presents an extension of the state-of-the-art work by tackling the problem of multi-lingual text detection and script identification.

In Chapter 4 we present our system based on the multi-level connected component (CC) analysis and the learning of text component features via convolutional neural networks (CNN). We then apply a graph-based grouping of overlapping text boxes. The system is evaluated on two databases of scene and born-digital images. It is found to be highly precise in detecting the text components and it outperforms state-of-the-art methods for the same datasets.

In Chapter 5 we present a new system for joint text detection and script identification

within an integrated methodology. Our system proceeds the tasks by casting the text detection task as an object detection problem, where the object is defined as the text of a specific script.

Chapter 2

Background and Relevant Literature

Contents

2.1 Introduction

Even though the popularity of text in image gives rise to several applications in many fields, the ultimate goal is to confirm whether or not text exists in a given image. Then, if the text exists, the goal is to detect, localize, define its script, the language used and recognize it. Through the various works of the literature, we noted similarities on the stages followed to finally recognize the text in the image. The various stages of those principal tasks mentioned in different works are: First, text localization which aims to localize the initial text candidates in the image. Then text detection which decides if there is a text in the image or not and define the text bounding box using the localization and verification procedures. Second, many works focus on text information extraction. This stage focus on both localization and binarization tasks. The goal is to rectify the distorted text in the image, improve the resolution and the quality of text strokes also called text enhancement. Third, other works focus on the detected text, such as script identification, language identification and text recognition. All these stages are essential for an End-to-End text detection and recognition system.

The stages mentioned above were initially executed on scanned pages images from documents. Then, with the evolution of the means of image capturing, researches on document analysis and recognition were extended. It included text detection and recognition of images captured with mobile camera and especially for text in the wild. Text appears everywhere in our environments. It is the written form of human languages and is among the most influential creations of humankind. The advantage of using the text is its ability to effectively and reliably spread or acquire information across time and space. It has become a vital tool for collaboration and communication. Therefore, automatic text detection and recognition from captured images in natural environments has become an important research topic rapidly growing in computer vision. The progress made in recent years in the topic of text analysis in the image can be summarized as follows:

- The incorporation of deep learning had a great impact in the domain of text detection (and many other fields). The efforts dedicated before deep learning for designing and testing hand-crafted features are now used for new tasks and to solve new challenges. Moreover, the use of deep learning simplified the overall pipeline (from stepwise to integrated systems) and present a significant improvement.

- New tasks are executed using the new challenge-oriented datasets with many spe-

cific aspects. Newly published datasets are collected with representative characteristics, for example, dataset of curved text [7], long text [8] or for blurred text [9], etc.

- New tasks and challenges emerged in the text analysis field. With the presence of new datasets and models devoted to the main task. Such as working on multilingual and multi-script text or different types of images with different characteristics (natural scene text and synthetic text) etc.

Despite years of research and many progress, many challenges may still be encountered when working on text detection in the image such as:

- The text in natural scenes exhibits much higher diversity and variability than text in the document. For example, text of scene images are often in several languages and scripts can have different shapes and sizes.
- The complexity and interference of backgrounds with the text. The problem is that the backgrounds in natural scenes are virtually unpredictable. There might be occlusions caused by objects in the background that lead to confusion and mistakes in the text detection and eventually the text recognition task.
- Another problem faced in almost all the natural scene images is the imperfect imaging conditions. Capturing conditions can add more challenges for the text detection task if it result in low resolution, severe distortions due to inappropriate shooting angle or distance or bad lighting, etc.

These difficulties run through the years before deep learning has shown the improvements it can achieve and the potential it has in computer vision and other research fields. The proposed methods are now performing on increasingly challenging goals and higher accuracy scores.

In the next sections of this chapter we will present an overview of the recent development in the text detection field from scene images in particular. We review methods from various categories: methods based on deep learning techniques and methods before the onset of deep learning. We also list the up-to-date datasets used for text analysis tasks in images. In our work, we mainly focus on the text detection and localization in some specific types of images, however, through our work we found a new description for the text that permit to detect the multilingual text and identify its script. Therefore in this chapter, we review some works focusing on the script identification task.

2.2 Classical Text Detection Approaches

In this section, we take a look at algorithms that predated the era of deep learning. In this period, text detection systems focus on different challenges than these faced now when working with the deep learning techniques. The used datasets in traditional or classical methods were not big enough to train deep models. Traditional methods are usually step-wise and with detailed pipeline, where each step is performed in a separate block. These methods focus on the local features and set the rules to extract and classify the text components. The attention has been focused on the design of features. There have been several excellent journal papers which organize and present comprehensive coverage of these works, such as the one of Uchida Seiichi [10]. In his survey he introduced the problems faced in text detection and recognition in images and video, then review the solution technologies to the realization of camera-based OCR. The review focus on three tasks: text image acquisition, text localization and text recognition. Then he details the existing methods to solve each problem faced while working on these tasks. The second survey [11] is written by Ye and Doermann. It reviews the methodologies in complete text detection and recognition systems: step-wise and integrated. It presents the fundamental sub-problems like text localization and verification, then text segmentation, text recognition and finally text enhancement. In their work, they also introduce the problem of multi-oriented text and multi-lingual content and many others. We also found two other interesting surveys about the traditional text detection approaches [12] and [13] which also categorize and analyze works related to text detection and recognition.

2.2.1 Specifications of Scene Text Images

To review the existing methods used to localize the text in an image, we first need to understand the specifications of the scene text image. Scene text images have special characteristics such as different colors, patterns, sizes, fonts, lighting conditions (shadows, darks, etc.), distorted perception, irregular backgrounds, lack of resolution, misalignment, uneven surface, etc. Figure 1.4 illustrates some of the difficulties faced. Among these difficulties we find:

- The complexity of the image background complicates the segmentation methods, especially for scene text. Text can appear in the scene even on an uneven surface. Perspective distortion, even moderate distortion can cause OCR to fail.
- Lighting conditions are controlled to a lesser extent by the camera built into the

cell phone (shadows, reflections, etc.).

- When taking a photo with a mobile device, the subject may be in motion. This navigation makes motion blur.
- Perceptual distortion : When the text plane is not parallel to the picture plane, cameras are subject to perceptual distortion. The distance between the text and the camera can vary, resulting in text with variable size.
- Low resolution: When the text image is far from the camera, image magnification becomes a solution. However, this may result in poor text quality.
- Font and size: The presence of multi-scale methods has not yet solved the problem of wide variation in text size. The text must be readable at a certain spatial and temporal distance. Text fonts in landscape photos are very diverse. This diversity of lines makes the recognition phase more difficult.
- Text direction: Most of the proposed methods assume that the text orientation is horizontal or vertical. Only few works focus on multi-oriented text.

2.2.2 Technical Approaches

Text localization should detect the text in the background of the image. Since the text of the image can be on a more or less textured background, with more or less contrast, it is necessary to develop strategies for detecting and preprocessing the text before submitting it to the recognition step. Several methods for extracting text information from images have been proposed for applications such as document segmentation, address block location, license plates identification and indexing of images based on content. Despite the works already done in this regard, it is still difficult to design textual information extraction in a general purpose system.

In fact, there are many difficulties when extracting text on a complex background from low-contrast images or on images where font size, style, color, orientation, alignment, etc. is changing. These differences make the problem of automatically extracting text information very difficult. The text extraction system from an image is generally divided into three stages. Figure 2.1 illustrate the different stages and the frequently used techniques for each stage.

- Firstly, from the input image they start by finding the initial candidates or regions of interest. This could be done at pixel or zone levels using sliding window, con-

nected components or local region based. Usually, this stage is the most challenging and involves many complicated preprocessing steps (illumination corrections, blur and focus corrections, filtering and noise removal edge enhancements, segmentation, etc.), it results in many candidates or regions of interest. The lost candidates in this stage are hardly found later.

- Secondly, the filtering stage(s) where initial candidates are classified as text or non-text components. Some methods use hand-designed features such as texture, color, orientation or shape that require many parameters and fine-tuning or feature detector such as MSER, SIFT, SURF to generate the features of the extracted regions. Then, the classification step uses either heuristic rules, morphological operations, or machine learning approach such as support vector machines or neural networks for classifying the text components.

- The third step is the grouping step, in which text components are grouped into characters, words or text-lines. Depending on the initial cutting level of components, one or more steps could be required in the grouping step.

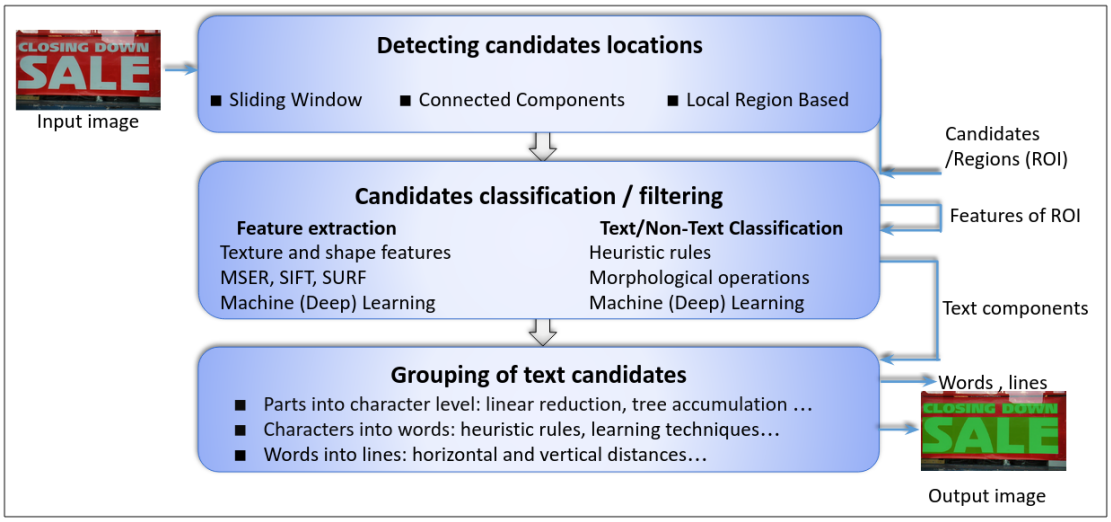

Figure 2.1: Typical pipeline of a classical text detection system. Three main stages can be identified. Samples of used techniques in each stage are presented with input and output data.

Depending on the used technique in each step, we can categorize the methods to three categories. The categorization depends on how to use the extracted features. They can be used by the machine learning methods. Heuristic rules can also be applied on these features in order to find a better representative model of the extracted text regions. They are also used by hybrid techniques that combine machine learning methods and heuristic rules.

2.2.2.1 Heuristic-based methods

They apply a set of manually crafted rules directly on the raw extracted features. Many methods in this category exploit text edge properties to extract the initial text candidates. Such as distribution, density and stroke width. These characteristics are then manipulated as sets of labeled features grouped by rules of similarities, for example color features, texture features, strong and symmetric gradients, size and fonts have been always considered as relevant features of texts regarding the background. Below we review some heuristic-based approaches:

Phan et al. [14] proposed a text detection method for video based on Laplacian operator. Maximum gradient difference value is computed for each pixel in the Laplacian-filtered image. K-means is then used to classify all the pixels into two clusters: text and non-text. Then for each candidate text region the corresponding region in the Sobel edge map of the input image undergoes projection profile analysis to determine the boundary of the text blocks. At the end, like all heuristic-based methods, they employ empirical rules to eliminate false positives and improve the final results.

Epshtein et al. [15] proposed a text descriptor named the stroke width transform (SWT) which quantifies the stroke width for every image pixel. The operator computes for every pixel the width of the foremost likely stroke that contains the pixel. By measuring the width variance of every component, texts are often easily extracted since text is characterized by fixed stroke width. This descriptor is widely used in text detection systems in order to improve the strokes of the text components in the image.

Huang et al. [16] proposed an edge-based method, called edge-ray filter, to detect the scene character. The main contribution of the proposed method lies in filtering out complex backgrounds by fully utilizing the essential spatial layout of edges instead of the assumption of straight text line. Edges are extracted by a combination of Canny and Edge Preserving Smoothing Filter (EPSF). Then an Edge Quasi-Connectivity Analysis (EQCA) is applied to boost the filtering step. Then non-characters are filtered using

Label Histogram Analysis (LHA). At the end they apply two frequently used heuristic rules, namely aspect ratio and occupation to improve the final results.

The steps followed in these methods are present in almost all heuristic-based approaches. Many steps are proposed for improving and extracting significant features from the image. Then heuristic rules are applied to filter the components and generate the final result.

2.2.2.2 Machine learning-based methods

Many machine learning-based methods are proposed for text detection. They aim to learn discriminating features from training data in order to create a text/non-text classifier. For text localization, these approaches usually use a window technique. The classifier scans the image at different positions and scales producing text candidates. A grouping algorithm is then applied to supply text regions. Basically, the introduction of machine learning was first done in the second stage of the system (Figure 2.1), which necessitate many handcrafted rules and computations in the heuristic-based approaches. Then deep learning has settled down and controlled more steps in the text detection system. This category will be reviewed in detail in the next section (Section 2.3). However, below we review some methods based on machine learning:

Lee et al. [17] proposed a text detection system in natural scenes. It is based on the Modest Adaboost algorithm, short for Adaptive Boosting, which is a statistical classification meta-algorithm formulated by Yoav Freund and Robert Schapire. In their work they used nodes of Classification And Regression Tree (CART), a non-parametric decision tree that determines outcome variables from among an out-sized number of features. Finally, they optimize the detected bounding boxes by combining the resulting maps at 16 spatial scales with equal weights into a single 1024×768 map. Then they derive an edge map from color gradients and use it as a criterion of region optimization.

Yine et al. [18] presented a learning framework based on a unified distance metric in order to carry out an adaptive hierarchical clustering. This approach can simultaneously learn similarity weights (to adaptively combine different feature similarities) and the clustering threshold (to automatically determine the number of clusters). Then, they constructed text candidates by grouping characters based on this adaptive clustering. They also tested their approach for the multi-oriented text.

Coates et al. [19] applied an unsupervised feature learning algorithm to a set of image patches harvested from the training data to learn the features of the image. A set of

feature vectors are learned using k-means algorithm. The learned features are used to describe sliding windows that sweep input images where a character/non character classification is performed using a learned linear SVM. They also trained a linear classifier for the character recognition task.

2.2.2.3 Hybrid methods

Hybrid methods are groups of methods that use both heuristics and machine learning. Generally, a rough text detection is first performed using heuristic rules. Then, a feedback pass takes place to reject false alarms using machine learning-based techniques. The resulting schema can then take advantage from both approaches and reduce time and complexity.

Huang et al.[20] combined the high capability of convolutional neural network (CNN) which is capable of learning high-level features to robustly identify text components from text, with a set of low-level heuristic features extracted using both MSERs and sliding-window based methods. The MSER was used to reduce the number of scanned windows and to improve the detection of poor quality text. While the sliding-window with CNN is applied to correctly separate the connections of multiple characters in components. This combination has led to good results against a number of extreme text variations.

Shivakumara et al.[21] used Laplacian and Sobel operators to enhance low contrast text pixels in input video frames. They applied then a bayesian classifier to classify true text pixels from the enhanced text matrix (without assuming a priori probability about the input frame but estimating it based on three probable matrices). Using boundary growing method based on the nearest neighbor concept, the method was tested on multi-oriented texts in video frames.

Another hybrid method for text detection in natural scene is presented by Zhao et al.[22]. It is made out of two phases. In a hierarchical stage, a learning-based on Partial Differential Equations (PDEs) framework is first learned off-line with L1-standard regularization on preparing pictures. Utilizing an extra nearby binarization step on the generated certainty maps, text locale applicants are then separated. In the base up stage, character competitors are distinguished dependent on their shading likeness and afterward assembled into text competitors by basic rules.

The methods of detection and localization of text that exist in the literature can also be classified according to the process of partitioning the input image into multiple

segments used in the first step in the text detection system: extraction of regions of interest (Figure 2.1). We find segmentation free and segmentation based methods. In segmentation-based approaches, the image is directly divided into regions. The goal of the segmentation process is to change the characteristics of the image into more meaningful ones, thus facilitating interpretation and classification. They start from the bottom up to identify small structures and combines them to form areas. Therefore, they are also known as "bottom-up" methods. While in segmentation-free based approaches, they generally exploit texture and learn textual properties. They believe that the text in the images has distinctive composite properties that distinguish them from the background. In this category, we can find methods using techniques based on Gabor filters, wavelet, fast Fourier transform, spatial contrast, etc. They can be used to discover the composite properties of a text area in an image.

2.3 Text Detection Systems Based on Deep Learning techniques

In this section, we detail recent advances such as changes in methodology and methods design. The methods of recent years present the following two distinctions. First, most of the methods use models based on deep learning. Text detection is now seen from different angles and methods are trying to solve new challenges. Methods based on deep learning have the advantage that learning of features can allow them to avoid designing and testing a large number of potential literal features. At the same time, researchers from different viewpoints are enriching and encouraging the community to do more in-depth work, aiming to achieve different goals, for example a faster and simpler pipeline.

Integrating deep learning was a game changer. The incorporation of deep learning has completely changed the way researchers tackle a task and has greatly expanded the scope of research. This is the most significant change from the previous era. The evolution of the architecture of the developed methods is illustrated in Figure 2.2. In this figure are presented pipelines of several popular works on scene detection based on deep learning networks. From top to bottom:

- (a) Horizontal word detection and recognition pipeline proposed by Jaderberg et al.[23]. His approach is a combination of region proposal methods that extracts many words bounding box proposals then filtered using a CNN. A second CNN is trained to recognize the words. Then the results are merged according to their

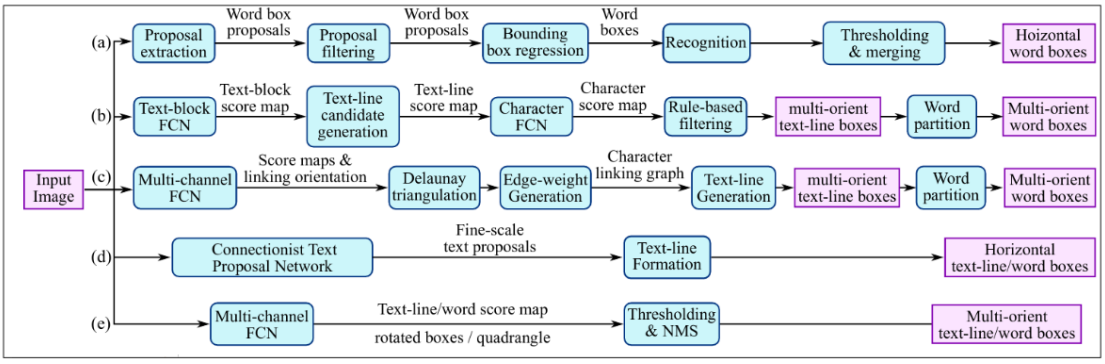

Figure 2.2: Typical pipelines of text detection systems based on deep learning. Figure taken from [1].

neighboring relations.

- (b) Multi-orient text detection pipeline proposed by Zhang et al.[24]. It takes into account global and local cues for localizing text lines in a coarse-to-fine procedure. Then, they train a fully connected network FCN for predicting the salient map of the text regions. Followed by a set of hypotheses combining the salient map and MSER components to define the text lines. Then a second FCN classifier is used for predicting the text characters.

- (c) Multi-orient text detection pipeline proposed by Yao et al. [25]. It consists of multiple stages and components, such as false positive removal by post filtering, candidate aggregation, line formation and word partition.

- (d) Horizontal text detection using CTPN, proposed by Tian et al. [26]. They slide the image and fed it through the last convolutional maps of the VGG16 model. The sequential windows are recurrently connected by a Bi-directional LSTM, where the convolutional feature of each window is used as input of the 256D BLSTM. The RNN layer is connected to a 512D fully-connected layer, followed by the output layer, which jointly predicts text/non-text scores, y-axis coordinates and side-refinement offsets of k anchors.

- (e) An Efficient and Accurate Scene Text Detector EAST by Zhou et al. [1]. They use a fully connected network FCN for feature learning and text classification followed by a Non-Maximum-Suppression NMS merging stage to form the final results. The network is designed for two geometry shapes for text regions: rotated box and quadrangles. They also designed different loss functions for each geometry.

The multitude of stages and components may require exhaustive tuning, leading to sub-optimal performance, and add to processing time of the whole pipeline but these systems perform well on many public datasets. The goal of researchers is not only to outperform the state of the art methods but also to ameliorate the architecture of their systems and to lighten the pipeline of their methods. The use of deep learning has decreased the complexity of the methods and improved the final results. Methods that utilize deep learning based models have the advantage to use and re-use pre-trained models. Fine-tuning or transfer learning save researchers from designing new architectures and training them from scratch.

2.3.1 Methodology in Deep Learning

In fact, many scene text detection algorithms are primarily inspired by and follow the designs of general object detectors systems. However, scene text detection presents a different set of characteristics and challenges that require unique methodologies and solutions. Existing CNN-based methods can be broadly categorized into several sub-classes: region proposal-based methods, segmentation-based methods and hybrid methods using multitask learning.

So the evolution of scene text detection algorithms goes through three main stages. First, the use of learning-based methods equipped with multi-step pipelines but these methods are still slow and complex. Then, object detection systems have been successfully adapted to the text detection task. The third point is that researchers design special representations based on the subtext components to solve the problems of long, curved and irregular texts.

2.3.1.1 Multi-step Pipelines with Deep Learning

The first approaches based on deep learning process the detection task into a multi-step process. Early deep-learning-based methods have used CNNs to extract the initial regions of interest, and then apply rules and post-processing steps to integrate these regions into significant blocks (words, lines or paragraphs) [27, 28]. Then CNNS have been used to classify the regions of interest into text and non-text components. And the extraction of the regions of interest was performed using MSER, sliding window, connected component extraction, etc [20, 29]. By applying CNNs in different steps, methods have been adapted to be applied to the whole image in a fully convolutional approach. It was used to predict whether each pixel belongs to a text block or not, or to identify the segmentation map

of the image indicating the text line regions, etc. We find text detection systems based on a unified neural network such as [30, 31].

The common point between all these methods is the design methodology. They are all designed in a bottom-up strategy and based on key components such as single characters or text center lines. Overall they are robust and outperform traditional text detection methods, but still have relatively long and slow pipelines unlike some object detection methods.

2.3.1.2 Text Detection Inspired from Object Detection

Text detection is a special case of object detection, as text can be thought of as a special type of object, and can be solved by an approach to object detection and/or recognition. Looking at the history of text detection research, it is strongly intertwined with the development of object recognition, sharing a similar evolutionary path and a similar paradigm from proposals from the pre-deep learning era to the deep learning era. Research on scene text detection has started treating words and characters as objects. Text detection systems are then designed by modifying the region proposal and bounding box regression modules of general detectors to localize the text instances. [32, 33, 34]. More details are presented in our work in Section 5.2.

2.3.1.3 Holistic and Sub-text Components Methods

The major difference between text detection and general object identification is that text is homogeneous as a whole and is defined by its locality, whereas general object detection is defined by its location. The characteristic that any section of a text instance is still text is referred to as homogeneity and locality. We do not need to view the entire text instance to recognize it as part of a text. A small part, defined as a character or more, is enough to identify that it is part of text. This characteristic is the foundation for a new branch of text detection algorithms that only anticipate sub-text components and then group them together to form a text instance. These solutions are more adaptable to the aforementioned problems of curved, lengthy, and oriented text. These approaches make use of neural networks to forecast local qualities or segments, as well as a number of other techniques. Sub-text components can be one of three levels: pixel-level, component-level and character-level methods.

- In pixel-level methods, fully convolutional neural networks are trained to generate a dense prediction map. This map will indicate whether each pixel in the original

image is text (part of text) or not. Wang et al. [35] predicted the text regions starting from the pixel-level. They applied different shrinkage scales and enlarged the detected text region round-by-round until a collision with another instance. Similarly Tian et al. [36] defined a loss term to maximize the euclidean distances between pixels from different instances, and so differentiate adjacent text blocks.

- Component-level methods predict the components at a medium granularity level. A component refers to a region of the text instance, it can be a word or group of characters. More details and used techniques are detailed in Section 4.2.

- Character-level representation requires datasets with ground truth labeled at character level, which is very rare. The proposed method will learn the text features from the segmentation map of characters. Baek et al. [37] proposed this idea in their work. The network learns a segmentation map of character centers and links them in the form of Gaussian heat map.

2.4 Review of Script Identification techniques

Benefits of using multi-lingual and eventually multi-script text include the creation and appreciation of cultural awareness, add academic and educational value, enhance creativity, adjustment in society and appreciation of foreign languages. Multi-script text nowadays exists everywhere in many countries, and multi-script text detection and recognition has become a new challenging task. There are many multilingual countries in the world where there is a text in road signs, advertisements, etc. It can be written in two or three languages. In Tunisia for example, we can find texts written in Arabic, French and English. In China, we always find the text written in Chinese and English. Thus, for automatic text reading, it is necessary to select the language before performing the text recognition step. Moreover, in many countries like India, many languages are used in different regions of the country. Only a few studies have offered analyses of multilingual texts with prominent visual aspects, such as advertisements, posters and web pages. However, before processing the analysis and recognition of this type of text, we first need to identify its script and language. In this case, we need to choose the appropriate language to get the best interpretation of the characters.

2.4.1 Different Levels for Script Identification Task

Script identification refers to an action in a particular document that defines the text of the written text segments. For comprehensive and recent surveys we find the one of Ubul et al [38] and the second one written by Parul Sahare and Sanjay B.Dhok [39]. Script identification is considered as a text classification problem that is solved using different statistical methods and different computational methods. In the literature, there are several methods for identifying scene scripts. Script identification methods can be classified into three categories according to the type of document to be analyzed. We find methods for printed documents, handwritten documents and others for video frames and captured images. In this section we review script identification methods in video frames and camera based images.

The script identification task has not been investigated much up until this point. As opposed to printed or handwritten documents, techniques for script identification in images initially require data detection and extraction. In published and handwritten documents, textbook in black appears generally on a simple background. Still, script recognition in video and camera captured images originates from complex conditions and suffer from low resolution, blur, complex background, noise, exposure problems, different textual styles and text dimensions of video text. All these complications make this problem more delicate and grueling than published and handwriting document identification. Some approaches on script information from video frames or camera based images at different levels are reviewed in the following.

2.4.1.1 Text Block Level

Gllavata and Freisleben [40] presented an approach for discriminating between Latin and Ideographic script. Their method, first, localize the text in the image. Then, extract a set of low-level features from the localized text image, basically texture features. Finally, based on the extracted features, the decision about the type of the script is made using a k-nearest neighbor classifier. Another interesting work for multi-script text detection and script identification is proposed by zhao et al. [41]. They collected their dataset and proposed a Spatial-Gradient based Features (SGF) at block level for identifying the text script among the six considered ones: Arabic, Chinese, English, Japanese, Korean and Tamil. The input for script identification step is the text blocks obtained in the text frame classification step. They first compute horizontal and vertical gradient information to enhance the contrast of the text pixels. Then divide the horizontal gradient block into

two equal parts, where the upper values present dominant text pixels. The method combines the horizontal and the vertical dominant pixels to obtain text components. Finally, used skeleton concept and adjacency information to identify the script of the text.

2.4.1.2 Text Line Level

Working at text line level is used when the output for text detection task is text line. Phan et al. [42] applied video script identification before choosing the appropriate OCR engine. The input data used is the text lines obtained by their text detection method. They proposed two features, namely smoothness and cursiveness, for video script identification at text-line level. For evaluating their system, they tested on combinations of two languages from three different scripts: English, Chinese and Tamil. Gomez et al. [43] proposed a method that combines convolutional features and the Naive-Bayes Nearest Neighbor classifier. Their proposed framework exploits the discriminative power of small stroke-parts, in a fine-grained classification framework. They also introduced MLe2e [44] multi-script dataset for the evaluation of scene text end-to-end reading systems and all intermediate stages: text detection, script identification and text recognition.

2.4.1.3 Word and Character Level

At this level, the data is presented as cropped words or characters. Sharma et al. [45] used traditional document analysis techniques to identify overlapping text scripts in video clips and analysed the text at word level. They analyzed three different combinations: Gabor filters, Zernike moments, and manually generated gradient features. They proposed a number of preprocessing algorithms to overcome the challenges inherent in text overlay video. They show that the combination of ultra-high resolution, color gamut properties, and a SVM classifier (Support Vector Machine) performs better than other combinations. Another work presented by Li et al. [46] reported a script identification based on statistical features technique to identify, at character level, English, Arabic and Chinese scripts of camera-based images.

2.4.2 Features For Script Identification

Feature extraction is a mandatory step of any recognition system. In the last years, different kinds of features have been estimated for script identification grounded on the characters of each script. Two broad categories of features have been established in the script identification fields: Local features and global features.

2.4.2.1 Local features

Local features are extracted from small textual candidates of the document image. For example in [47], they blended medium level representations and deep properties into a globally trainable deep model. The authors extracted deep local features in each Multi-Stage Pooling Network 'MSPN' layer and described the images with code-based encoding. They described the images using a codebook-based coding method that can be exploited to adjust CNN weights. Local features mainly consider the analysis of intrinsic features such as character shape based features, structural features, statistical features, water reservoir principle based features, morphological, topological and contour based features, etc. The extraction of these features is time consuming, but they convey relevant characteristics for script identification.

- **Statistical Features** present mathematical characteristics such as the mean and variance of the width, height, ratio and area of the connected components. Methods that analyse the upward concavity, vertical and horizontal projections, cooccurrence matrices [48] etc. use this type of features for identifying the script. Statistics-based approaches are highly sensitive to noise and image quality. Some of the commonly used statistical features are: vertical and horizontal projections, Water reservoir-based features, bounding box feature, upward concavities.
- **Template Matching Features** have the advantage of differentiating similar scripts. This type is very sensitive to the variations in font and size of the characters in the same script. It superpose an unknown pattern to the ideal template pattern and the degree of correlation between the two is used for classification.
- **Structure and Geometric Features** such as loops, cusps, endpoints, starting points, etc.. They depend on the instinctive aspects of the writing and are based on the geometric appearance of scripts. Some typical structural features are: Heuristic features, Fractal-based features, topological features and morphological features.

2.4.2.2 Global features

Global features are extracted from blocks of text of the document image. These features are robust to noise, small skew, and faster in computation than local features. In general, they are considered to be efficient in characterizing large size texture patterns, such as text blocks. Gabor Filter, Wavelet Transform features, gray level co-occurrence matrix, rotation invariant features, gradient features, Texture-and Steerable pyramid-based features are global features often used in the literature. However, as these features regard

a text block as one whole entity, the analysis at certain levels such as: word, character or connected component, is not possible. [40] [41] and other works have used this type of features.

2.5 Datasets for Scene Text Detection

One of the main challenges in the field of information retrieval is the data collection and data manipulation. In many computer vision fields, data present the backbone of text analysis systems from the detection step to the optical character recognition systems development. Indeed, in order to build a process able to provide good results, it is necessary to provide the classifiers with the required data (quantity and quality) to extract significant features. For many decades, emphasis has been mainly given to scanned documents and handwritten documents (historical documents). Embedded text on photos or videos has received lower attention. Moreover, the most important public datasets in terms of diversity and size are usually restricted to some languages (specially Latin script). The lack is remarkable for some scripts and languages. And also for some possible tasks (the required ground truth is not provided).

One dataset can be used for many tasks, depending on the ground truth annotation and the number of images it contains. In Table 2.1 we reviewed some of the existing public datasets. All the available datasets related to multi-script text detection and to script/language identification are listed first. For each dataset we mentioned its key information. First, the source of the images: captured natural scene image outdoor or indoor, Google street view images or captured from digital images (television videos, web advertisements, etc). The second parameter is the number of scripts. Most datasets with more than two scripts are listed in this table. We also mentioned the well known RRC (Robust Reading Competition) dataset for scene text detection (FST: Focused Scene Text and BDI : Born-Digital Images) as they are used in most text detection works. The third parameter is the number of the images and the distribution used for the training and the testing phases. We also precise the type of the provided images: full image or only cropped text. Finally, we precise the tasks that can be executed on the dataset. The different tasks are:

- Text or Multi-script text detection / localization
- Text segmentation
- Script and language identification on full or cropped image

- Joint text detection and script identification
- Text recognition
- End-to-End text detection and recognition

Table 2.1: State-of-the-art datasets for multi-script scene text detection and/or script classification

Dataset/Year	Images Source	Script/Languages	Images/and Type	Tasks
SIW-10 [49] (Script Identification in the Wild) Year: 2015	Google street view images	10 scripts: Arabic, Chinese, English, Hebrew, Greek, Korean, Russian, Thai, Tibetan, Japanese,	Cropped text images Train: 8045, Test: 5000	Script classification (>500 per script)
SIW-13 [50] (Script Identification in the Wild) Year: 2016	Google street view images	13 scripts: Arabic, Chinese, English, Hebrew, Greek, Japanese, Korean, Russian, Thai, Tibetan, Kannada, Mongolian, Cambodian	Cropped text images Total: Train: 9791, Test: 6500	Script classification
CVSI-2015 Video-text Dataset[51] (ICDAR-2015 competition) Year: 2015	Extracted from Television videos (such as news and advertisements)	10 Indian scripts: Hindi, Bengali, Oriya, Gujarati, Punjabi, Kannada, Tamil, Telugu + English and Arabic	Cropped text images Total: Train: 9791, Test: 6500	4 tasks: all tasks are script classification from subsets of the dataset scripts, or all of them
KAIST [52]] Year: 2011	Outdoors and indoors scenes	2 scripts: Korean and English	3000 complete images	1.Text localization 2.Text segmentation

Dataset /Year	Images Source	Script Languages	Images/ and Type	Task
ILST: Indian Language Scene Text [53] Year: 2016	Natural scenes: (1) manual capture from street scene in various Indian cities (2) Google image search (3) importing from other existing datasets	6 scripts: Telugu, Tamil, Malayalam, Kannada, Hindi and English	500 scene images with more than 3000 words (>500 per script)	1.Cropped word script identification 2.text localization with script identification (end-to-end pipeline)
MLe2e [44] (multi-lingual end-to-end) Year: 2016	Scene images from various existing scene text datasets: KAIST,MSRA-TD500,Chars74K, MSRRC	4 scripts: Latin,Hangul Chinese, Kannada,	711 images, all resized to: 640x480 which is the image size of the KAIST	1. Multi-lingual scene text end-to-end system 2. Text detection 3. Script identification 4. Text recognition
Multi-script Robust Reading Competition [54] (MRRC) - ICDAR 2013 Year: 2013	Multi-script text from scenic images obtained from Indian roads	2 scripts: Kannada and English	334 complete scene images (167 for training, 167 testing)	1.Multi-lingual 2.Text segmentation 3.English word recognition 4.Kannada word recognition 3, 4: from cropped word images
MSRA-TD500 [55] Year: 2011	Outdoors and indoors scenes	2 scripts: Chinese and English	500 complete scene images	Text Detection (multi-oriented)
FST [3]] Year: 2015	Outdoors and indoors scenes	Latin script	462 complete images	1.Text localization 2.Text segmentation 3.Word recognition 4.End to End
Born-Digital [3] Year: 2015	Captured from the web	Latin script	551 complete images	1.Text localization 2.Text segmentation 3.Word recognition 4.End to End

2.6 Conclusion

To understand the general value of text detection and image analysis approaches, it's useful to supply background information about the underlying problems faced in this field, technical challenges and review the appropriate methods used. In this chapter we presented a review of the existing techniques for text detection in the wild. The existing methods could be classified into two categories: methods before the area of deep learning and methodologies based on deep learning. Then we give a summary of the popular methods used for the script identification task. Finally, we presented a review of the existing datasets used for the text detection task and explain how the performance evaluation is done.

This Chapter explored the use of neural networks learning techniques to improve the performance of text detection systems, and the classical pipeline for a hand-crafted rules based system. We choose to make use of the deep neural networks to solve some of the faced problems in this field and to achieve better results than state-of-the art methods. Given the important number of systems and methods to localize the text in the image using the deep learning techniques, we had to choose the targets on which we are going to focus.

Therefore, we had to identify the major hurdles on the reviewed methods in Sections 2.2 and 2.3. They will present the main focus in our proposed methods.

- The use of deep networks and complex systems (number of processing and post processing steps on the image) is acceptable as it permits to achieve good results. But what if we can achieve the same results with less steps. With a better **alleviated pipeline** that minimize the number of steps but value each step to be the most efficient.

- Another key point we noticed in the existing approaches is the use of big datasets, many works even take advantage of data augmentation and train the network with different data from synthetic and natural images. However, we can focus on **the quality of the extracted components** from the image and try to generate valuable inputs to the network.

- Deep neural networks actually represent the last link in the evolution of artificial intelligence and allow researchers to learn the best features. However, we find text

detection systems trained and tested on specific text type (language or script), only few works treat the text as object and work on the **multi-lingual text**.

- Also deep learning systems use a large amount of resources, including memory to store models and data, and computations to process the different data, resulting in high power consumption and considerable computation time. Smaller network but with **better settings and fine-tuning** achieves the same results or better and with less needs in memory and data.

Chapter 3

Multi-Lingual Text Database

Contents

While working on text detection and script identification approaches and after a deep study of existing databases, we note the need to build a new dataset. As in all computer vision fields, data presents the backbone of the data analysis systems. The datasets available in the literature are mostly not multilingual. In this chapter, we propose a new dataset composed of complete scene images. It offers interesting novel aspects and

challenges. This dataset has been used for the ICDAR 2019 Robust Reading Challenge on Multi-lingual scene text detection and recognition.

3.1 Introduction

Datasets are fundamental to foster the development of several computational fields, giving scope, robustness and confidence to results. Many scientific research are based on the gathering and analysis of data. Text detection and recognition in a natural environment is also essential to many applications such as tourist guidance, helping the visually impaired, data mining and autonomous driving, etc. With the rising of deep learning, datasets are becoming more important and can be seen as the primary intellectual output of a research. Especially, when the data is reproduced and will be used in the future for longitudinal research, the training or testing of a method.

The datasets available in the literature for scene text detection are mostly not multilingual. As we can see in Section 2.5, the datasets which contain multi-script text are either built for Indian scripts only, or they contain a small number of scripts (2 - 4) with a relatively small number of images. Moreover, datasets that have been created for the tasks of script identification (classification) for example are composed of cropped text word images. Therefore, we got the need to collect large sets of images. And we focused on complete scene images. A second part of the dataset is synthetic that matches the real dataset in terms of scripts. This synthetic dataset was added to help with the training for the end-to-end text detection and recognition task.

In the next sections of this chapter, we will present the different steps to build the dataset and how to prepare the data to be used for different tasks. We started by the data collection then the data annotation. Within every step, we focused on checking and proofing the data and the ground truth annotation.

3.2 Data Collection

The data collection process is the first step for building the dataset. Multi-Lingual Text MLT dataset is composed of 20000 captured scene images and 277 000 synthetically generated images with the same set of languages to assist the training. The synthetic dataset matches the real dataset in terms of scripts and it is provided to help with the training for the End-to-End text detection and recognition.

Real Scene Images:

- Type and source of images: The images are natural scene images with embedded text, such as street signs, street advertisement boards, shops names, passing vehicles... The images were captured using different mobile phone cameras or were collected from freely available images from the Internet. The images mainly contain intentional focused scene text, however, some unintentional text may appear in some images. Such very small, blurry and/or occluded text.

- Number of images, Languages and Scripts: The dataset is comprised of 20,000 images containing text of 10 different languages (2,000 images per language). Most images contain text of more than one language, but each language is represented in at least 2,000 images. The ten languages are: Arabic, Bangla, Chinese, Devanagari, English, French, German, Italian, Japanese and Korean. Those languages belong to one of the following seven scripts: Arabic, Bangla, Chinese, Hindi, Japanese, Korean and Latin. An eighth script class named "Symbols" was added for special characters. And a "Mixed" script is used when characters of two or more scripts appear in the same word (without spaces).

Synthetic Scene Images:

Hence, we have provided an additional synthetic dataset [56] to complement the real one for training purposes. We adapt the framework proposed by Gupta et al. [57] to a multi-language setup. The framework generates realistic images by overlaying synthetic text over existing natural background images and it accounts for 3D scene geometry. The generated dataset contains the same set of script classes as the real dataset: Arabic, Bangla, Chinese, Devanagari, Latin, Japanese and Korean. Examples from each script are presented in Figure 3.1. The Synthetic Multi-Language in Natural Scene Dataset contains text rendered over natural scene images selected from the set of 8, 000 background images collected by [57]. Annotations include word level and character level text bounding boxes along with the corresponding transcription and language class. The dataset has 277000 images with thousands of images for each language.

Figure 3.1: Example of synthetic text from the different scripts on scene images.

3.2.1 Acquisition Process

The first step in building the dataset is to set up data collection and labeling tools. In order to collect an important number of images per script (2000 images per script), we asked colleagues, friends and family to capture natural scene images whenever they can. Therefore the images are taken under real life situations without caring about the environmental conditions and refined camera settings and on different sizes. People who have participated in the collection of the images are from different countries. For each script we have prepared a group of persons who accepted to send us the images regularly until we achieve the required number of images.

We have imposed rules for the collection of the images of our dataset related to the type, contents and capture conditions.

This is to ensure homogeneity of the collected images of the different scripts. And to make our dataset a meaningful benchmark providing the ability to differentiate the different scripts using the characteristics of the scripts and not the characteristics and patterns of the pictures containing the scripts.

For this dataset, we choose to capture natural scene images indoor and outdoor with

embedded text taken in daylight or artificial light relatively bright, such as shops names, street signs and advertisement boards, passing vehicles etc. Dark or night images, even if they contain text, were not accepted (they present an additional challenge for the users).

In Figure 3.2 and Figure 3.3, we present a sample image from each script. This kind of images represents the mostly encountered image types. It is also the most popular on the internet, we also downloaded some images from the internet (with the acceptance of the owners and copyrights).

Figure 3.2: Examples of scene images with embedded text of different languages

During the data collection process the number of images per script was fixed to 2000 images. Each script set has been then divided equally in two subset for one for training and one for testing.

Here we refer to full scene images, each of them may contain more than one word.

Figure 3.3: Examples of scene images with embedded text of different languages

Many images contain one or more scripts. We classify the image in the script it contains the most. For example, Latin and Arabic scripts coexist in many Arabic images. But as the Arabic script dominates in the image, it will be classified in the Arabic script category. The same problem faced with Chinese and Latin script also.

3.2.2 Data Quality Check

Along with data collection, we proceed a quality check of the received images. The goal of this step is to ensure the homogeneity and the quality of the used images. In this step, few points must be checked.

Some of them have been executed manually, but some scripts have been developed to do it. For example: images with offensive content or personal data have been systematically deleted. Captured images may contain some handwritten text, when too much handwritten text appear in the image it will also be removed. As handwritten text detection and recognition follow different techniques, only few cases are allowed in the dataset. Also each scene image must contain at least one word of the language we are collecting it for (example: Hindi images must contain at least one Hindi word, not only English words). And if it contains text from other languages not selected in the dataset, then this text will be ignored. Moreover, some preprocessing steps are executed on the

image for security issues such as blurring the images that have license plates of cars or people faces (except advertisement models or political figures).

A second phase of proofing and checking is performed after the ground truth (GT) annotation . The goal of this step is to check the ground truth annotation and prepare the images to be uploaded. After uploading the files of the different scripts, a list of algorithms are performed automatically to generate the list of files containing errors, and precise them.

The general rules for the ground truth annotation and the structure of the GT files are listed in Section 3.3.2. At this step, it is important to check and recheck all the annotated images. The cropped word images for the script identification task are also prepared at this stage.

3.3 Database Labeling

This step is as important as the data collection step. The importance of the dataset and its use depends on the quality of the labeling process. It also depends on the amount of information extracted from the images and the consistency of this data. For the MLT dataset, we used the RRC platform for the annotation and the generation of the ground truth files at xml format.

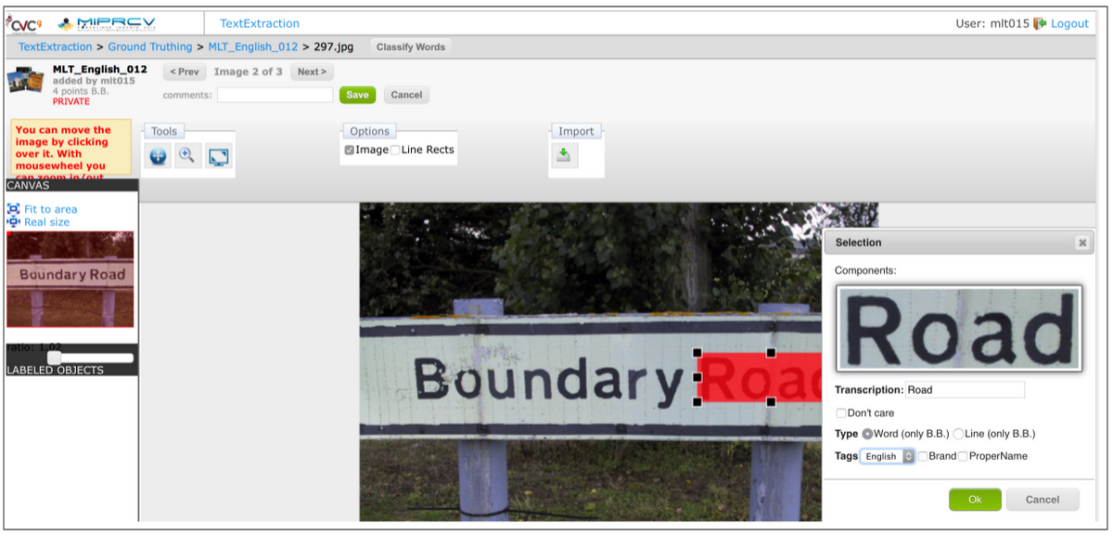

Figure 3.4: Example of labeling an image with the RRC platform

The text in the scene images of the dataset is annotated at word level. A GT-word is defined as a consecutive set of characters without spaces, i.e. words are separated by spaces, except in Chinese and Japanese where the text is labeled at line level. Each GT-word is labeled by a 4-corner bounding box, and is associated with a script class and an Unicode transcription of that GT-word. Some text regions in the images are not readable to the annotators due to low resolution and/or other distortions. Such regions are marked as "don't care" and ignored in the evaluation process. Figure 3.4 shows the platform used for the annotation process.

3.3.1 Key Features for Annotation

Our main rule is to type what we see exactly in the scene without making an extra effort to guess the text or the language. We focused in this part on the rules followed for the Latin and Arabic script. Most of the rules explained below in Table 3.1 are valid for all the scripts and languages. Also in "don't care" cases, there are two cases: either the language tag of the text is clear and should be assigned but the transcription is not clear, or the tag is left empty if the text is not clear (blurry or very small).

Table 3.1: Examples of the annotation rules used in the MLT dataset

Type	Example	Annotation
Numbers	0,1,2,3 ..... should be always assigned as English when they appear as a separate word (even if the context of the image (the other text, even within the same sentence) is another language – languages of Latin script or otherwise).	Script:Latin Language: English
	separate series of digits (it could be one digit) is considered a word: example "123456" is a word and its transcription is "123456"	Script: Latin
	"123 456" are two words and the transcriptions are: "123" and "456"	Language: English
	Arabic numbers used in the middle east	Script: Arabic Language: Arabic

Type	Example	Annotation
Real numbers	"23.45" this is one word, transcription "23.45"	Script: Latin Language: English
	"50km" is one word, transcription "50km"	Language: based on the context
	"30 km" is two words, transcriptions are "30", English and "km", language based on context	Script: Latin
Words	assign the language id according to both the word and the context of the sentence or the part of the scene in which the word appears.	
	"normal" alphabet characters with special characters (symbols, punctuation etc.) within the same word	Same script / language of the alphabet
	two languages within the same sequence of characters without space.	Script: Mixed
Brands	the word takes the same language of the sentence or the part of the scene in which it exists (context).	
	For Latin-script languages, If the brand-name word exists by itself in the scene, it is labeled as English, this makes it an English image.	
	any brand name of any language take the tag indicating that it is a brand name, example if the word SAMSUNG in Korean language, should be assigned the Script:"Korean" and "Brand"	
Proper names	assign the word as the language of the sentence or the part of the scene in which it exists (context).	
	For Latin-script languages, If the proper name exists by itself in the scene, it is labeled as English, this makes it an English image.	
	For Latin-script languages, If the proper name word exists by itself in the scene, assign it as English, this makes it an English image.	
Symbols:	Symbols and punctuation characters.	Script: Symbols
	Non-readable text has the transcription,blurry / too small text or if some characters of a word are occluded or faded "xxxx",	Script: don't care
	For Latin-script languages, If the proper name word exists by itself in the scene, assign it as English, this makes it an English image.	

3.3.2 Ground Truth Structure

Images annotation was performed using the Robust Reading Competition 'RRC' platform. This tool automatically creates the xml files of the required format. Many open source tools exist for images annotation such as Labelme, Labelimg or CVAT which is also used for video annotation. For building this dataset, we used the RRC platform for some scripts such as the Latin and Arabic scripts and other local tools for others. At the end each image has two ground truth files: text file 'txt' and extensible markup language 'xml' format which present the native RRC annotation platform format .

The text file is a space separated file, where each line corresponds to one word in the image. The number of lines in the file is the number of words in the image. The ground truth file has the following format:
x1 y1 x2 y2 x3 y3 x4 y4 "transcription word 1" s l b p
x1 y1 x2 y2 x3 y3 x4 y4 "transcription word 2" s l b p
Where **x1, y1, x2, y2, x3, y3, x4, y4** are integer numbers that represent the coordinates of the 4 corners of the minimum bounding box that surrounds each word.
s is the script id. It is a number in [0,6] or one of 101, 103, 107, detailed in Table 3.4.

Table 3.4: The ten scripts and their identifier. Note that mixed, symbols and any ambiguous combination are defined as scripts.

Identifier	Script
0	Arabic
1	Bangla
2	Chinese
3	Latin
4	Japanese
5	Korean
6	Hindi
101	Symbols
103	Mixed
107	Other (for scripts not in the list above)

l is the language id. It is a number between 0 and 9 or one of 101, 103, 107, detailed in Table 3.5.

- **b** is a number either 0 or 1 that represents if the word is a brand name (1) or not (0).

Table 3.5: The ten languages and their identifier

Identifier	Language
0	Arabic
1	Bangla
2	Chinese
3	English
4	French
5	German
6	Italian
7	Japanese
8	Korean
9	Hindi
101	Symbols
103	Mixed
107	Other (for scripts not in the list above)

- **p** is a number either 0 or 1 that represents if the word is a proper name (1) or not (0).

- The values of **"b p"** could be: "0 0" or "0 1" or "1 0" or "1 1"(Both are 1 if a brand is also a person name).

'Mixed' class groups all the words that contain a group of characters from different scripts linked without any space in between. It can be a group of letters for example from Arabic and Latin script, or Arabic word linked to a number. As numbers belong to Latin script then we consider it as Mixed script. For the 'Other' or 'Don't care' class it will contain the words that are unreadable or belong to other scripts which are not considered in this dataset (this is rare, the majority are non-readable words). Also punctuation and some math symbols sometimes appear as separate words, those words are assigned a special script class called 'Symbols'. This ground truth structure is provided for all the dataset, then depending on the task to be executed, another format with the required data is provided (explanation of the ground truth files per task is provided in Section 3.4).

Before proceeding to the data preparation for each possible task, we recheck the annotated data. First thing to check is the quality of the bounding box around the word. We ensure to have the smaller and most precise bounding box around the word. We also check all the other data. In some cases the choice is not evident for example for the language id. Some languages of the Latin script share the same words (example:

communication, information, taxi ...), in this case we check the other words or the whole context in the image.

3.4 Database Organization

The annotated elements in the image are all saved in the ground truth file. During the annotation process, we put all the possible information extracted from the image. Starting with the bounding boxes and the text transcription, script and language, up to additional information such as specifying whether it is a brand or proper name. From these data we can run several challenges. These challenges have been proposed during the *ICDAR 2019 Robust Reading Challenge on Multi-lingual scene text detection and recognition* that we have organized. Each challenge or task requires specific ground truth details for the training phase. For testing the methods, only test images are provided. Unless otherwise stated, the prior knowledge for any of the following tasks is only the training images for all the scripts which appear in our dataset. To detail the organization of the dataset, we will go through the different possible tasks (or challenges) proposed during the competition and based on the MLT dataset. As we mentioned before the generated (original) ground truth files contain all the extracted data. This data is then reorganized according to the specific task.

Among the different possible tasks, four tasks are prepared and presented later as challenges to which participants have tested their methods. For all the challenges, only the ground truth for the training data is provided. For the testing phase, only images are provided to the participants.

3.4.1 Multi-Script Text Detection Task

This task might be seen as a notion of the primer spotting task of the precedent robust reading competitions, where a developed method should detect the text of many scripts once. Dealing with multi-script text is more challenging, as it requires more training and robust feature learning systems.
The input data is the complete scene image. One image may contain any script from the considered scripts in the dataset. It can have words from one or more scripts. The ground truth files are comma separated files, where each line corresponds to a single

text box in the image. It provides the bounding box coordinates (four corners clockwise) followed by the script id within this format:

x1, y1, x2, y2, x3, y3, x4, y4, script, transcription

The transcription field is not necessary for this task, however, it is added so the same ground truth files could be used for many challenges once.
The output data are bounding boxes of the detected words in the image regardless of their script or language followed by their confidence score. The results should be saved in UTF-8 encoded text file per image. Each line in the file corresponds to one word in the image.

3.4.2 Cropped Word Script / Language Identification Task

This task works on the cropped word images. All the other tasks are executed on the full image. There are many languages of the text in our images that share the same script such as : French, English, German and Italian from Latin script. Therefore we provide two tasks: Script identification and Language identification. However, once the language is known then the script is systematically assigned to the language. For this task, only 8 different scripts are considered as we have excluded the words that have 'Mixed' script and 'don't care' class.
The input data is the cropped words in all the dataset presented as separate image files, along with the corresponding ground truth script / language and transcription. The transcription is not needed for this task and can be ignored. The ground truth is presented in a single text file for the whole collection, where each line presents one cropped image and has the following format:

word image name, script, transcription

or

word image name, language, transcription

For text verification and context understanding in case of confusion about the script or the language, we provide information about the original image from which the word image has been extracted. In this file are provided the relative coordinates of the bounding boxes that present the text block within the cut-out text block image for the whole collection. Each line in the file presents one word in the image and has the following format:

word image name, x1, y1, x2, y2, x3, y3, x4, y4, (original image name)

The output data is presented in a single text file. Each line in the file present one image, and indicates the identified script / language among the proposed scripts / languages following this format:

word image name, script

or

word image name, language

3.4.3 Joint Text Detection and Script Identification

This task presents the last step before the End-to-End multilingual text detection and recognition. All the preparation steps needed for the multi-script text recognition are combined in this task. The difference with the previous task is that the script identification is performed on the full image.

The input data is a complete scene image with the same ground truth file provided for the first challenge.

The output data are the bounding boxes of the detected words followed with the script id. One file per image contains the coordinates of the different bounding boxes with confidence value and detected script following this format:

x1, y1, x2, y2, x3, y3, x4, y4, confidence, script

3.4.4 End-to-End Text Detection and Recognition

This is a very challenging task of a unified OCR for multiple-languages. When working on English script, the end-to-end scene text detection and recognition task setting is coherent. However, when dealing with multi-script and multi-language text, new difficulties might appear. Given an input scene image, the target is to localize a group of bounding boxes and their corresponding transcriptions. For this task additional data for the training is provided. This data is a synthetic dataset that matches the real dataset in terms of scripts and languages. The goal of this dataset is to help the training for this task.

the input data is complete scene image with the same ground truth file provided for the first challenge. Plus, the synthetic dataset and its ground truth files.

The output data should be provided as one text file per image. Inside each text file, a list of the detected bounding boxes followed by the transcription of the detected text in

his format:

x1, y1, x2, y2, x3, y3, x4, y4, confidence, transcription

3.5 Evaluation Metrics

In the competition, different evaluation metrics are used depending on the executed task. The proposed evaluation tools can be tested online or offline, on the RRC platform or locally. The evaluation metrics used for all the tasks in this competition are standard (classical metrics). In particular, the metrics used for Task 1 (Task 1: Multi-Script Text Detection 3.4.1) follows those of RRC [9] (which are also used in Wang et al. [58] and Everingham et al. [59]).

For the **Multi-Script Text Detection Task** (cf. section 3.4.1) an identification is considered as right (true positive) if the detected bounding box has over 50% cross-over (intersection over union) with the ground truth box, disregarding the case. The F-measure (Hmean) is utilized as the measurement for positioning the members strategies. The standard f-measure depends on both the recall and precision of the detected word bounding boxes when contrasted with the ground truth. Moreover, for the missing or 'don't care 'detected words marked as '####', they are not taken into account and will not affect the final results (neither positively nor negatively). Any detection covering over half with '####' ground truth locales will be disposed of from the submitted results before evaluation happens, and the evaluation tool will not consider ground truth transcriptions set as '####'. In light of these counted true or false identifications, the standard recall, precision and f1-measure (H-mean) will be determined for every participant methods.

With respect to **Cropped Word Script / Language Identification Task** (cf. section 3.4.2), the multi-class arrangement precision is utilized as a measurement (disarray grid). Where the precision is determined as the class-weighted normal of true identification across every one of the pictures (across all contents or language classes). This is a norm broadly utilized measurement for assessing characterization execution. The evaluation of results against the ground truth is computed in the following manner: participants give a script ID to each word picture, and assuming the outcome is right, the count of right outcomes is increased. The final accuracy for a given strategy is the accuracy of such prediction. This can be summed up with the straightforward definition that follows:

Let G = g1 , g2 , . . . , gi, . . . , gm be the set of correct script classes in the ground truth, and T = t1 , t2, . . . , ti , . . . , tm be the set of script classes returned by a given strategy, where gi and ti allude to the same original picture. A script identification score per word is considered right (One) whenever gi = ti, in any case it is false (Zero), the amount of all m IDs separated by m gives the overall accuracy for this task.

For **joint text detection and script identification Task** (cf. section 3.4.3) we will utilize the two measurements H-mean and characterization precision clarified in the other tasks. Thus, every strategy will have two outcomes during evaluation. A result is considered true if it presents correct text localization (detection of the correct bounding box) and correct script identification (script of the word). Joint text detection and script identification is counted as correct if the word bounding box is correctly detected according to the evaluation criteria of the text detection task and also according to the evaluation criteria of the script identification task.

Finally, for the evaluation of **End-to-End Text Detection and Recognition** (cf. section 3.4.4) we consider the cascade of correct text detection result (precise bounding box for the detected word) and correct text recognition result (exact transcription of the detected word). In order to respect the homogeneity between the training data and the test set, all the words in the test data that contain characters which do not appear in the training set, will be set as 'don't care '. Whether these words are correctly detected and/or recognised or not they will not affect the final evaluation scores. This rule is considered to ensure that participants could train their models based on the lexicon of the training set we provide.

3.6 Discussion

Building this dataset was a complete challenge and full time work. Multi-Lingual Text dataset, as presented above, is among the rare multi-lingual datasets.It is composed of two types of images. On the first hand, captured natural scene images: 20000 images from 10 languages belonging to 7 different scripts. Each language is represented by 2000 images equally divided between train and test set. On the other hand, 277000 synthetically generated images where the embedded text belongs to the same scripts and languages as the first part. We presented the different steps to finally get the full dataset labeled and ready to be used.

This Chapter has also presented different challenges of the "ICDAR 2019 Robust Reading Challenge on Multi-lingual scene text detection and recognition" using the MLT dataset. This competition has been carried out in the framework of the "Robust Reading Competition" (https://rrc.cvc.uab.es/) which gathers different research challenges on the topic of robust reading. The framework provides (among other things) an infrastructure for managing users and submissions, online evaluation and visualization tools. For the "ICDAR 2019 Robust Reading Challenge", there has been a total of 60 different submissions distributed over the four proposed tasks. This proofs a big interest of the community in the problem of multi-lingual scene text detection and recognition and the importance of the multi-lingual text dataset.

Chapter 4

Learning Text Component Features via Convolutional Neural Networks

Contents

Finding good initial regions of interest can greatly impact the final performance of a text detection system. Following a traditional pipeline of text detection systems starting with detecting RoI, learning the text features and finally grouping the results for word level, we build a novel method for detecting text in scene images.

4.1 Introduction

Picture-text congruence in our natural surrounding environments appears everywhere. In traffic signs, license plates, advertisement billboards, business cards, building signs, labels on posted parcels and on name plates, the text presents a valuable source of data. This data is then filtered and used in many applications. For example, in the tourist sector, these data are needed : for interactive tourists' guidance or providing text accessibility for visually impaired people (whether it is a necessity for their everyday life or simply for navigating or enjoying the world around them), or for potential applications of Web documents images analysis including commercial data mining (handling large volume of advertisement images), information extraction (extracting various kinds of text information from images) and Cyber security (searching sensitive content in images).

The text detection problem has been widely explored and many applications have been developed. These applications are usually focusing on a specific dataset and well fine-tuned in order to provide good results. Although it bears similarity to OCR problems in traditional document images, text detection is much more challenging in natural images. In addition to all the acquisition problems that may occur like the lighting, the shadow and occlusion problems,etc., the structure of the image could present a challenge. The complex layout with variable backgrounds and the high variations in text color, font, size and orientation affect the detection task. Moreover scene images usually contain more than one type of text, possibly touching or overlapping, along with background contextual noise. That's why the trend in computer vision has been to start by describing the content of images in terms of local image regions.

Building a system with local regions-based representations makes good use of the image characteristics since region-based representations have the advantages to be robust to occlusion, clutter and image transformation while extracting the information. This proves the importance of the choice of the initial local-regions or what is called Regions of Interest (RoI) that represent parts of object up to complete objects. In our case, it will be parts of the text up to complete word possibly with some additional noise from the background. Moreover, identifying good RoI is an important step because they present the only source of initial candidates for the system. If the RoI were consistently and precisely found, the latter steps in the system would give good results. Besides, it can greatly improve the performance of the object or text detection system.

In order to overcome the mentioned challenges, in this chapter, we propose a novel method for text detection in natural scene images. Our method is composed of three main stages.

- In the first stage, finding initial candidates is performed by multi-level connected component extraction (reasons for this choice will be detailed in the next section). This module handles complex background and variations in text scale/color by multiple binarizations. The module extracts redundant components of text/non-text at different granularity (text components could be parts of characters, characters, parts of words etc., and could be found multiple times). Our technique at this stage minimizes both the preprocessing steps, and the possibility of loosing potential text components before the next stages.

- Secondly, the filtering stage(s) where the initial candidates are classified as text or non-text components. For this stage, two ways are possible to follow: either to use hand-designed features and multiple filtering steps, or using deep learning methods. However, in the last years, the most marked trend adapted in the text detection systems is the transfer from traditional and conventional methods based on hand crafted rules to the deep learning methods. This is due to the multiple problems encountered with the hand-designed features and in order to achieve better results with standard systems. For this stage, we choose to build a convolutional neural network and to extract the features from raw components. The CNN is trained as a binary classifier to discriminate text from non-text components. Preparing the ground truth at connected component level was necessary in this stage. We build the network from scratch and set all the parameters of the network experimentally, including the architecture, the type and number of layers.

- Finally, for the third stage, we propose a general grouping method. The classified text components are first aggregated to form meaningful higher level components via linkage-based clustering (for example, broken parts of the same character would be grouped in the same cluster). Then, a graph is formed based on overlapping criteria of the components bounding boxes. The connected graph components form text words. These grouping steps do not pose assumptions related to the text script/language.

Many methods have been developed and accomplish the same goals. The work pro-

posed in this thesis presents many advantages. First it outperforms state-of-the-art methods when tested on the same datasets. Secondly, using simple techniques it achieved good results on different types of images. Moreover, the method might be simple, but it proves that a good choice for the input data to the network has a great impact on the results. In this chapter, we detail the steps of the proposed method and we review the different existing works using similar techniques.

4.2 Theoretical Considerations in Applying Connected Component Retrieval

Most works of scene text images and born-digital documents analysis focus on one or more of the following main issues: text detection and extraction, text recognition, layout analysis and symbol recognition. The problem of scene text detection and extraction in natural scenes is radically different from text segmentation in scanned paper documents. The latter three problems adopt some methods similar to those in scanned paper document analysis, but have also developed some specific methods. Document analysis systems always starts by text detection and extraction. The text localization and segmentation from scene images are generally considered more difficult than pure text recognition. The reason for that is the use of off-the-shelf OCR engines applied to the extracted binary text image after the segmentation task. The objective of the text localization task is to localize all the text components in the image precisely. And eventually to present them as final text component, by grouping them into text region candidates separated from the background as much as possible.

In order to have a global classification of the existing methods for detecting text candidates, we categorize them into two approaches: region-based approaches and holistic approaches. Our method, presented in this chapter, belongs to the first category. Region-based approaches are based on techniques detecting the initial regions of interest. They can be using either connected component analysis (CCA) or sliding window classification. Both are widely used in text detection systems. Color, edges and gradient, texture and other properties are typically used as features, some works use hybrid features.

A review of the techniques used in text detection systems for detecting the initial regions of interest is presented in Chapter 2. This section reviews only region-based ap-

proaches using connected component techniques.

Epshtein et al. [15] presented an image operator for text detection using the local value of stroke width at pixel level. The proposed Stroke Width Transform SWT combines dense estimation computed at every pixel with non-local scope as stroke width depends on information contained sometimes in very far apart pixels. They first apply SWT which results in an image where each pixel contains the width of the most likely stroke to belongs to. Then, they extract all the connected components (CC) in the image. The extracted CCs present letter candidates that will be later filtered. They apply these steps on the image and its complement in order to get more possible text candidates. While for example Huang et al. [60] used color information in addition to the SWT then they extracted the connected components. For the latter steps of filtering and grouping the possible text candidates, both works used a set of rules based on hand-crafted features.

In another work, Huang et al. [20] used both sliding windows and MSER in order to identify the initial regions of interest. The purpose of using MSER is to reduce the number of scanned windows. The same steps are followed by Wang et al. [61]. Both works apply deep CNN for classifying the extracted regions into text and non-text candidates. Zhang et al. [24] also extracted the character candidates using MSER for extracting initial regions of interest. Then a CNN is applied as a discriminative codebook to compute a bank of responses for each candidate. And the final decision for predicting whether a candidate is text or non-text is done using an SVM classifier.

The use of connected components might also be for improving the final results. In Zhu et al. [62], the authors first used convolutional k-means to learn feature banks. Then used confidence-rated AdaBoost to classify patches as foreground that presented text and background which presented non-text. Finally, they applied CC extraction from characters candidates using color and edge features to improve the word segmentation and change the output to word level.

Jiang et al. [63] presented a two-task network with integration of bottom and top cues. The first channel for labeling the image at pixel-by-pixel level. Based on that labeling word proposals are generated with a connected component analysis. The second channel aims to output the character candidates used in the later steps. Both sub-networks share the basic convolutional features.

Meenakumari et al. [64] used MSER regions for detecting the initial regions of interest. Their system detected the text and found the connected regions, chain them

together in their relative position. The result of multiple segmentation hypotheses are then post-processed by a connected component analysis algorithm. Xiaobing et al. [65] also extracted connected components from multiple segmentation. They used Markov Random Field with local contrast, colors and gradient of RGB channels. Then, they extracted connected components from each image channel. Then all the resulted components were filtered using two SVM classifiers. The remaining components considered as text components are then merged to generate the final detected text components. Finally, some rules are applied to finalize the grouping step and form word at level text.

As discussed in Section 2.2, and illustrated in Figure 5.1, text detection methods usually follow the schema of extracting initial regions of interest. Then they label them according to the original ground truth. The classification of the detected candidates is either done using heuristic-rules or convolutional neural networks. All the reviewed methods above grouped the detected text components according to the similarity in geometric and other heuristic properties such as stroke width, color, size, horizontal and vertical distances, etc. In the general run of things, working with a region-based approach required some pre or/and post processing manipulations.

In summary, working with region-based approach are usually related to the textual properties of the text to separate it from the background. The techniques mostly use wavelet, Gabor filters, spatial variance, etc. Working at the connected component (CC) level to find initial text candidates is preferred to the pixel-level or interest point level – which are noise-sensitive and slow –, and to the sliding window level which cannot be easily adapted to multiple scales and orientations among other problems. These methods further use machine learning techniques for example SVM, MLP and AdaBoost. These techniques work on the top down fashion as explained before. They started by extracting the texture features for later classification and finding the text regions then perform the grouping of the detected regions. This step is usually done based on some hand-crafted rules. Other post-processing steps might be required in order to improve the final result and present it as required at word or line level.

4.3 The Text Detection System

The region-based approach makes full use of the different region properties to extract the text objects. It utilizes the fact that there is always a relatively big difference between

the text components and the corresponding background.

This section describes the proposed method for text detection. The method is working on a bottom up fashion by first extracting the regions of interest using a multi-level connected component technique. Then, using a convolution neural network, discriminative features are learned from the raw components. To finally apply a graph-based grouping technique to aggregate the classified text components into words. Figure 4.1 shows the architecture of the proposed method.

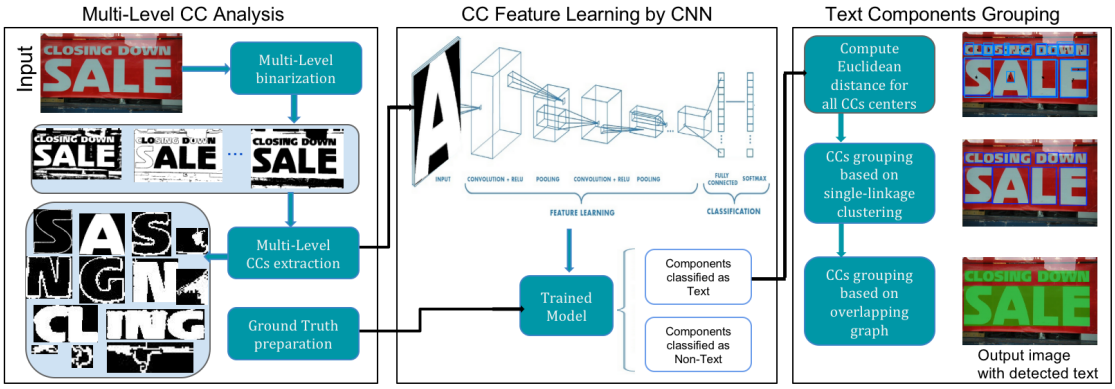

Figure 4.1: Block diagram of the proposed method with its three modules. First, Multi-level Connected Component (CC) Analysis based on multi-level binarization. Second, Feature Learning with the CNN from the raw multi-level CCs. Third, grouping of text components based on linkage clustering and overlapping graph.

The first module handle multi-level connected component extraction where a scene image is fed to this module as input. Multiple binarizations are applied on the input image and its complement (negative image) before extracting the connected components from each binary image. This ensures the extraction of low-contrast, light-on-dark and low resolution components. The resulting text and non-text components could be broken and/or extracted multiple times.

The second module is a classification module. It is composed of a CNN able to learn significant features from the raw components in the training phase. The generated model will be then used in the testing phase to separate text components from all the rest.

The third module aims at assembling the text candidates into words. Text candidates detected in the second module are grouped in two steps. First a linkage-based clustering

is applied to group broken and redundant components of the same character into the same cluster. Secondly, the clusters are merged based on the amount of overlap among the different bounding boxes of text.

Each module is presented in detail in the next Subsections, and the method is then evaluated in Section 4.4.

4.3.1 Multi-Level Connected Component Analysis

Working at the connected component level resulted in very precise candidates but also into many candidates. CC extraction relies so much on the lossy binarization step and may result in broken and noise components. Therefore, a good binarization for the image will greatly impact the quality of the extracted connected components. And here the choice of the adequate binarization for each image become an important step. Given the nature of the scene images and the superposition of the text in it, a single binarization step either adaptive or global cannot separate the foreground components from the background. As an example, scene images may contain light text on dark background in part of it and the opposite in the other part (as shown in the first image in Figure 4.2). In other cases, text in the image might be bold or just skeletons, linked or broken characters due to resolution and contrast variations. During binarization, some parts might appear using a specific binarization technique or might be lost using another. Moreover, the properties of images are quite different from a database to another.

How many binarizations to apply and which binarizations are better to get good connected components are two important factors to consider. The choice was validated experimentally, after testing different binarization methods, we chose 3 different approaches that we apply on the image and its complement one or more times with different thresholds. Each binarization is considered as a filter. Figure 4.2 shows the three different binarized versions of two example images. Each binarization reveals different components of the image. In our multi-level binarization, we perform three different types of binarization one or more times with different thresholds.

The first binarization method is inspired from the text localization approach of Chen et al. [66]. The idea of this work illustrated in Figure 4.3 is based on the fact that the text strokes have always complete contours and pixels on the contour have higher contrast than adjacent pixels. If we consider the image as many layers overlaid, then

Figure 4.2: Multi-level binarization results of two test images. From left-to-right images are shown followed by their binarization results for: smooth/non-smooth binarization, adaptive thresholding binarization on the original image and on the complement of Hue channel. Note that some text components appear in only one of the binary images.

the text component will be present in a separate layer than the background. So, for this binarization, to start with we separate the three color channels in the image and work on each color layer separately. First we compute the gradient magnitude per pixel in order to compute the local contrast and generate an image with the largest value of the gradient magnitude. Secondly, the image is segmented into two parts. One part contains smooth regions which are the pixels with low contrast. And the other part contains non-smooth regions which present pixels with high contrast. All the non-smooth regions are in our case considered as text regions. But we also extract connected components from the smooth regions which fill the non-smooth regions. At this level, we do not exclude any possible component. Finally, all the extracted smooth regions are merged with the non-smooth image to generate the final binarized image.

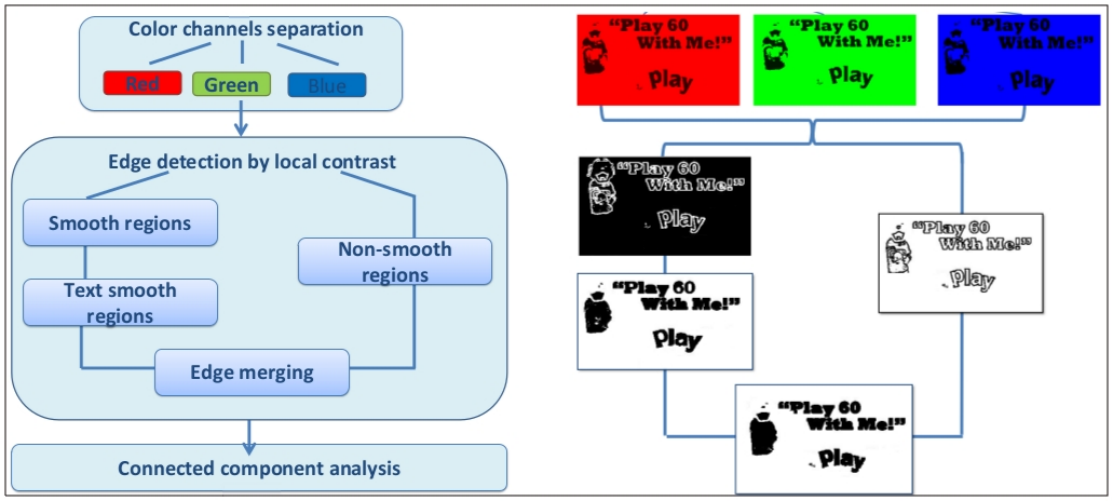

Figure 4.3: Block diagram of the proposed binarization technique. The different steps are applied on each color layer image separately. An example of the generated image on each step on the right part.

The second binarization is based on local adaptive thresholding. From the histogram of the image, an individual thresholding for each pixel is selected based on the range of intensity values within its local neighboring pixels. This operation is repeated until it converges. And the threshold is then computed by separating the histogram of intensity values into two classes.

The third binarization is computed from the HSV color space. The second binarization technique is applied on the complement of the Hue channel. The combination of these steps, allows to find low contrast components and light-on-dark components also.

4.3.1.1 Data Preparation at Connected Component Level

As a result of the multiple binarizations applied on the image, three different versions are saved. The step of extracting the connected components is applied on the different images. All the results are then saved. The extracted connected components are composed of text and non-text components. Text components could be a letter , a group of letters linked together or part of a letter. They can also be broken or extracted multiple time (the same connected component could be present in more than one version of the image). Extracting all the components present increase the probability of finding all text components, hence minimizing the loss of any possible text candidates at this early stage.

The redundancy in the extracted connected components and the broken parts are two problems that will be dealt with in the next modules.

Combining the components from all the binary images yields to many components of different types, but it would minimize the number of lost components. In Figures 4.4 and 4.5 are illustrated examples of the CCs extracted from the multi-level binarized images. Besides that they might be redundant and broken, those candidates are of different fonts, sizes, and backgrounds.

Figure 4.4: Samples of extracted CCs from different binarizations of different images. The components are of different sizes, orientations and shapes, and of variable types: letters, groups of letters and varying non-text components.

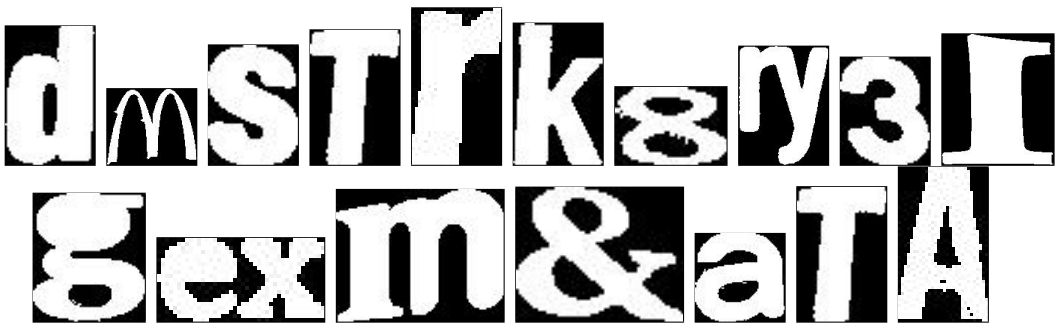

Figure 4.5: Samples of extracted connected components from different binarizations. Those are the most frequent type of extracted text components.

To a large extent, all the steps of multi-level binarization and multi-level CCs extraction aim at minimizing as much as possible the loss of any potential text component. At this stage, the text is defined as a connected component with a complete contour. No assumption of the text characteristics is considered, not even the script or orientation of the text is defined in advance.

4.3.1.2 Efficient Ground Truth Annotation

The remaining problem after extracting all the possible connected components is on deciding their labels. In the datasets of scene text detection, the ground truth is usually annotated at word or text-line level. Therefore, in order to prepare the input data for the training phase to the next module of the system, we prepare a ground truth at the connected component level. From the original ground truth annotation, we have the coordinates of the different text bounding boxes. A connected component is labeled as text if it overlaps with the text bounding box in the ground truth with a ratio higher than 0.8. Otherwise, it is labeled as non-text component. Figure 4.6 illustrates the process. The use of a high overlap ratio yields to accurate ground truth labeling of the candidate connected components.

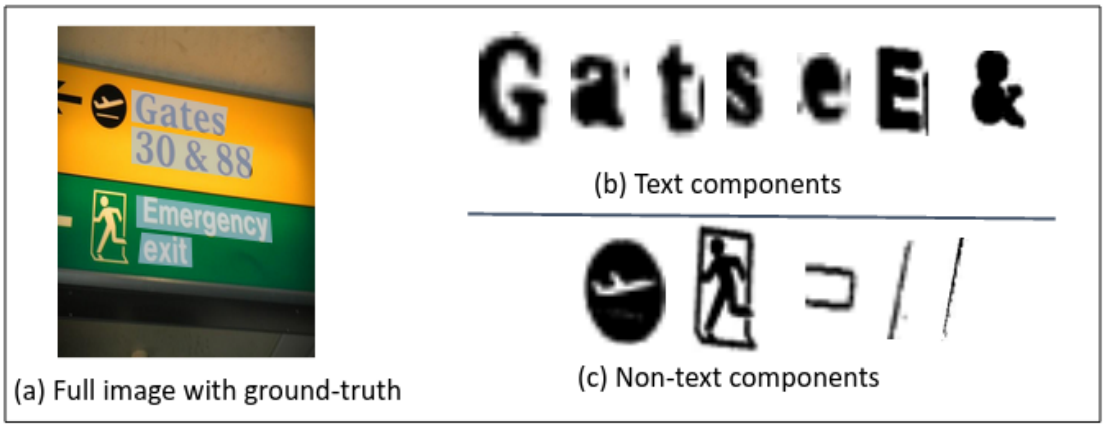

Figure 4.6: Ground truth labeling for connected component level. From (a) we apply the different binarizations and extract all the possible CCs. If the CC is included in a text bounding box from the ground-truth then it is in the group (b) else in the group (c).

The result of the multi-level CC extraction module are thousands of raw text and non-text components with variable content and size characteristics.

4.3.2 Learning Text Component Features via a CNN Classifier

The advantage of using the multi-level CCs extraction is that all the possible candidates in the image are extracted. Also these candidates come without any assumptions related to their properties. However this presents a challenge for the classification step. As the

task should be fast and provide good results. To compute the likelihood that a component is a text or not, we built a CNN classifier. The deep features of the components are learned using this CNN classifier. For that we used three types of layers illustrated in Figure 4.7:

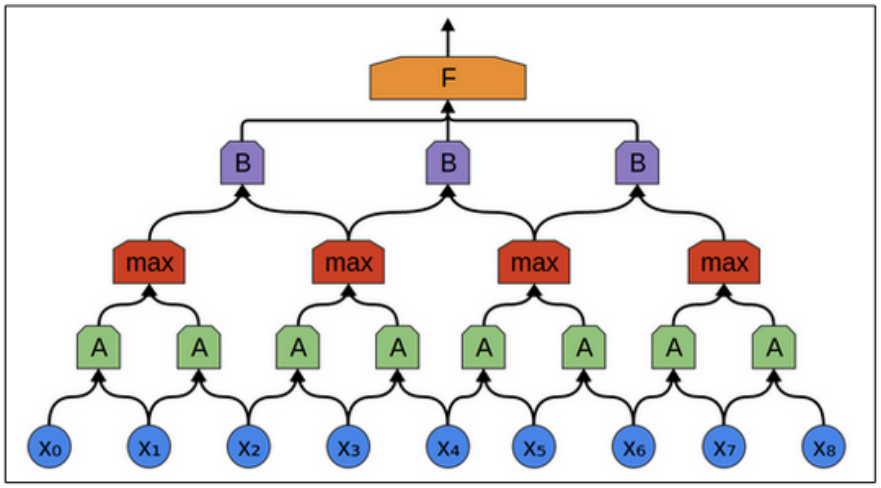

Figure 4.7: Abstract representation of the convolutional neural network. (X0,X8) present the data segments. *A* presents the convolution layer.*max* is for the max-pooling layer. *B* is used to create another convolutional layer stacked on top of the previous one. *F* for the fully-connected layer.

- Convolution layer: is a Deep Learning algorithm which can take in an input image, assign weights and biases to various objects in the image and is able to classify them into different classes. In Figure 4.7, *A* only looked at segments consisting of two points. This is not the case. The convolution layer's window is too much larger.

- Max-pooling layer: is responsible for reducing the spatial size of the convolved features. This is to decrease the computational power required to process the data through dimensional reduction. It is important for extracting dominant features which are rotational and positional invariant.

- Fully-connected layer: is used to learn non-linear combinations of the high-level features as represented by the output of the convolutional layer. The Fully-Connected layer is learning a possibly non-linear function in that space.

We also used Rectified linear unit after each convolution layer. The rectified linear activation function or ReLU for short is a piecewise linear function that outputs the input directly if it is positive, otherwise, it outputs zero. It is the default activation function for many types of neural networks as it result in models easier to train and often achieves better performance.

There exist many available networks to do this task, some works rely on the transfer learning from pre-trained models. In our case, we choose to build a relatively small network to learn the relevant features. All the parameters and the architecture of the network are set experimentally. Figure 4.8 shows the architecture of the used CNN. The proposed network starts with a data layer. All the enclosing boxes of the components are fed as separate input images to this data layer. All the input components are normalized in the data layer. And because of the small size of these components, we opted for small kernels.

After the data layer, there is a series of 4 convolutions followed each by a rectified linear unit layer and separated by two pooling layers. They are followed by a fully connected layer, where the last layer generate 1-D feature vector containing the final prediction of the concerned component. The classification result is binary and the used loss function is the standard *softmax* function.

Feature learning in CNN goes through two phases: training and testing phase. In the first phase, the data is composed of the images of the connected components and their corresponding ground truth. Different feature maps are generated from the text and the non-text component images at the different network layers. The model is trained for learning the deep features from the raw components and generate the training model. This model is used for the testing phase. We first apply the same steps of the multi-level connected component extraction on the test images. Then, they are used as input data to the generated model.

The output of this module is the prediction value for a connected component for being text or non-text. In the next module, only text components are considered. The rest of the components labeled as non-text are not used in the next step.

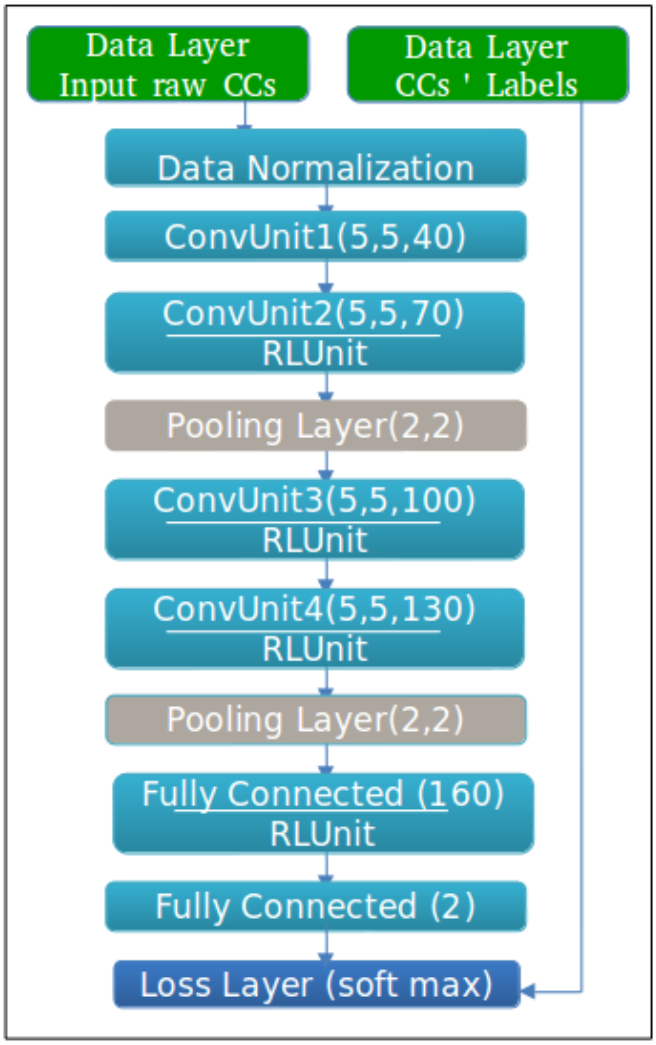

Figure 4.8: Structure of the CNN classifier network. The ConvUnit(w,h,n) represents a convolution layer of n features with wxh kernel size, connected to a ReLUnit layer and pooling layer with kernels of size 2x2. They are followed by two fully connected layers of 160 and 2 outputs respectively.

4.3.3 Graph-based Grouping of Text Components

The Last module in the text detection system is the grouping of the components to form the final output. The proposed grouping method takes as input the components labeled as text from the classifier in the previous module. The problem faced when working with CCs based methods is the absence of text line construction algorithms. Therefore, the grouping module could not take advantage of this information. In our method, we

do not focus on the text line orientation because we assume that the text in the image could have multi-orientations. And so we do not put assumptions of the eventual text orientation.

The grouping method must solve two main problems resulted from our multi-level connected component strategy which is the broken components and the redundancy. The grouping method consists in two steps. First, linkage-based clustering is used to aggregate the redundant and broken text components. Parts of the same text component will be aggregated in the same cluster. Redundancy problems will also be solved by laying the pieces on top of each other. The resulted cluster will group all the present details of the same text component. In the second step we build a graph based on the overlapping criteria of the components bounding boxes of each cluster. Basically, this step is for assembling the components formed from the first step into bigger blocks which are the final words. Figure 4.9 shows the described grouping steps applied on an image.

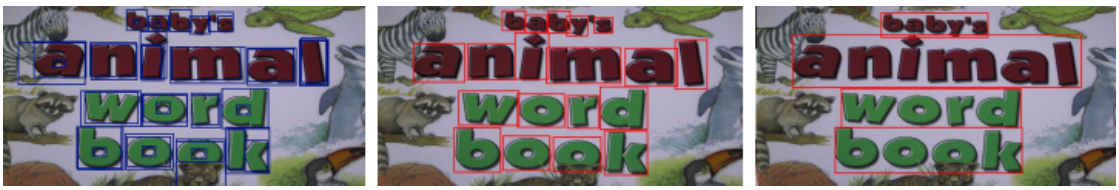

Figure 4.9: Left: original image with all the text components. Middle: output of the first grouping step: the resulting boxes are mostly letters (or few merged letters merged). Right: final grouping output: text grouped at word level.

To build the linkage-based clustering, we start by computing the euclidean distance matrix between the centers of each two text components. Then, we create a dendrogram based on a single-linkage hierarchical clustering. Here, the text component clusters are built in a bottom-up way where at each step, a pair of text components are putted in the same cluster if they are closest to each other based on the distance matrix. Then, more clusters are formed by merging smaller clusters together. This pair-wise merging process is repeated until no pairs could be further merged. These steps will force broken components of the same character to be merged in the same cluster, as well as redundant versions of the same text component. At the end of this step, each cluster will be presented as a single bounding box. These boxes present the input text candidates for the next step.

In this step, the bounding boxes of the text candidates are the nodes in the adjacency

graph. To create the edges, we proceed each pair of boxes at a time as follows. First we compute the overlap between the extended bounding boxes of each text component. Then we build an edge between these two boxes if their overlap is higher than the threshold. This threshold is adaptive to the scale of the boxes. We choose to work with the extended bounding boxes, because the extracted connected components are very precise and fit exactly the stroke of the text, and in the same word, a small blank can be found between letters. Finally, in the case of a successful grouping, the adjacent nodes in the resulted graph represent parts of the same word.

Following this strategy, we got the advantage to recover some missing parts in the text box. Extra small parts included in the text box but classified as non-text or not detected are merged into the final text box with these steps, such as the dots or missing parts of the text. Also by extending the size of the bounding box of a connected component with respect to its original size, the small components will be recovered if they have neighboring text components.

Finally, this grouping method also has the advantages over rule-based methods which necessity many hand-tuned parameters. Or in our case, it is based on hierarchical clustering of text components centers and it is adaptive to the scale of text components when computing the overlapping graph. Besides, like in the previous modules, no assumption related to the text properties such as script and orientation is made.

4.4 Experiments and Performance Evaluation

We have implemented the proposed methods in a text detection system. The input to the detection system is a full color image. The image might be captured or born-digital. We choose to evaluate our system on different types of images. We worked on relatively small datasets. The number of the non-text components extracted from the scene images is important compared to the text components number. To ensure the balance between text and non-text components, we choose to train our model with relatively small datasets. The implementation parameters and more details about the used datasets are presented in the next sections.

4.4.1 Implementation Details

Within the different modules of the system, there is no preprocessing or post-processing manipulation of the image. In the first module, the multi-level connected component extraction, we worked on the original image and did not proceed any modifications.

For learning the features from the multi-level raw connected components extracted from the images, we used the CNN network presented in Section 4.3.2. The CNN was trained by stochastic gradient descent with back-propagation and a maximum number of iterations of 10^4. The learning rate policy is *fixed* and the base learning rate is 10^{-3}. Weight decay is $5x10^{-4}$ and momentum is 0.9. We build the network with 4 convolution layers, with a fixed kernel size of 5x5 pixels and different number of features in each layer. For the first layer we used 40 outputs, for the second one 70 feature maps. The next two convolution layers have 100 and 130 feature maps respectively. The following fully connected convolution layer has 160 outputs, and the final layer has 2 outputs: text and non-text classes.

The experiments have been conducted using Caffe library [67]. Caffe provides a deep learning framework. It offers many models and optimization that are defined by the configuration and do not require hard-coding. Moreover, it is among the fastest ConvNet implementations and presents detailed documentation.

4.4.2 The Database

The system is evaluated on the standard public dataset of the ICDAR2013 Robust Reading Competition Challenge2: Focused Scene Text and Challenge1: Born Digital images [2].

Focused Scene Text (FST): This dataset contains images taken under real life situations with variations in the lighting, focus, and angles of the shots. It is composed of 462 images, split as 229 training images containing 848 Latin words and 233 test images. We used the same split that the one used in the competition setting.

Born-Digital Images (BDI): This dataset contains images synthesized on computers such as advertisements and Web images with embedded text in full size larger than 100x100 pixels. It is composed of 420 images, containing 3583 Latin words of more than 3 characters for the training set and 102 images, containing 918 words for the testing set.

4.4.3 Experimental Results

For the evaluation of the system, we used the standard recall, precision and f-measure metrics proposed in the RRC competition [2] and used in most scene text detection works. A detected bounding box is considered as a match if it overlaps a ground truth bounding box by more than 50%.

However, we started by evaluating the multi-level connected component extraction module. The number of inputs to the CNN network – the second module of our method – depends on the number of filters used for binarization in the multi-level CC extraction module. These inputs are the text and non-text extracted component images. In Table 4.1 we showed the impact of using different binarizations. Using three binarizations yields to getting more correct text components. The accuracy presented in Table 4.1 is for the classification of the different extracted components, without the grouping step. This experiment also shows the effectiveness of the CNN in learning good features despite the number of the non-text components. It also shows that the CNN aims to retain all possible text components and successfully classify parts-of-character components.

Table 4.1: The extracted text and non-text connected components of the RRC dataset [2] using multiple binarizations, and the resulting CNN classification accuracy.

Number of filters	Type	Text	Non-Text	Total	Accuracy
One binarization	Train	4920	4160	9080	-
	Test	4359	8100	12459	78.29%
Two binarizations	Train	5980	4553	10533	-
	Test	4465	8600	13065	87.97%
Three binarizations	Train	6221	8519	14740	-
	Test	5820	12289	18109	**96.81%**

Secondly, we show our text detection results compared to state-of-the-art text detection methods. Table 4.2 and Table 4.3 show the text detection results of our method applied on the RRC dataset Challenge1 and Challenge2 [2] respectively. Both tables show the effectiveness of the system compared to other works evaluated on the same datasets. Many newer works could be found working on the same datasets, but our method is still an efficient system for detecting the text in such type of images. Using simple techniques and well designed small network, we achieved good results without resorting to big CNN, pretrained model to be fine-tuned, or tricky preprocessing steps.

Table 4.2: Text detection results of the proposed method compared to state-of-the-art methods on the Challenge1 RRC dataset [2]

Method	Recall	Precision	F-measure
Our method	94.87	97.21	**96.02%**
Wu et al.[68]	91.00	95.00	93.00%
Villamizar et al.[69]	88.00	94.00	91.00%
Nayef et al.[70]	88.38	91.87	90.09%

Figure 4.10 show qualitative results of our method on focused scene images. The detected text is very precise. The text in natural scene images is always clear and words are well spaced. The results are in most cases at word level. As shown in the figure, text with different sizes is also detected.

Figure 4.10: Examples of successful text detection results on focused scene test images [2]. The detected text is shown in green bounding boxes. The detected regions are mostly precise and cover a word or a text-line.

Table 4.3: Text detection results of the proposed method compared to state-of-the-art methods on the Challenge2 RRC dataset [2]

Method	Recall(%)	Precision(%)	F-measure(%)
Our method	82.28	89.94	**85.94%**
Zhu & Uchida [71]	84.00	83.00	84.00%
Zhang et al. [72]	88.00	78.00	83.00%
He et al. [73]	73.00	93.00	82.00%
Faster R-CNN [74]	75.00	86.00	80.00%
R-FCN [75]	76.00	90.00	83.00%

Figures 4.10 show qualitative results of our method on born-digital images. The final detected regions are in most cases precisely detected. And adjacent text is also detected without missing components. The results are at word level. However, in few cases it might be at line level. This is the case when the space between two words, horizontally or vertically, is very close to the space between characters in the same word.

Figure 4.11: Examples of successful text detection results on born-digital test images [2]. The detected text is shown in green bounding boxes. The detected regions are mostly precise and cover a word or a textline.

4.5 Discussion

This chapter has described a text detection system applied for the scene and born-digital images. It combines multi-level connected components extraction with the use of CNN-based feature learning for components classification.

The method has few limitations when dealing with some special cases. Figure 4.12 shows some failure cases. Due to the similarity of some logos to text letters, some logos are miss-classified as text. Also, bad acquisition conditions affect the quality of the extracted connected components and so the final detected text. As an example, the bad lighting conditions or the very low resolution are the most two problems present in the focused scene image dataset.

The problem faced in this work is that when working with connected components, we extract all the possible patterns in the image, which is a good point. However, there are

Figure 4.12: Examples of failure cases: strong highlights, transparent or very small text. Red boxes show missed text, green boxes show correctly detected text.

many patterns which appear to be characters. In other words, we faced many ambiguous and confusing shapes in scene. For example, on the corner of a room or a plate, we will find 'Y 'shaped edges. Around leaves of trees, we also find dense and fine edge structures which can be classified as dense text lines. Reciprocally, some decorated characters seem like a branch of a tree. This difficulty, let us ask again, what are the character patters? More precisely, how to differentiate between character pattern and non-character pattern when they are similar? Based on these questions, we found that to make a difference, we should take advantage of the neighboring relation between all the extracted components (not only in the grouping stage, like we did in this work) in all the steps and consider the context of the image. Starting from the extraction of the initial regions of interest to the build of the final word bounding box. All these issues have been analyzed in order to define the second system presented in Chapter 5.

Chapter 5

Deep Neural Network for Joint Text Detection and Script Identification

Contents

The text detection system presented in Chapter 4 follows a step-wise approach. It finds most of the text in the image and performed well on different public datasets. Using multi-level binarization before connected component extraction, and learning the feature components with a CNN is a good combination for filtering text components. Even if we did not use hand-crafted rules within the three main blocks of the system, we still can minimize the number of needed steps to get the final result. In this chapter we present a new system for joint text detection and script identification within an integrated

methodology. Our system proceeds the tasks by casting the text detection task as an object detection problem, where the object is defined as the text of a specific script.

5.1 Introduction

Nowadays, we acknowledge a huge growth in the amount of multimedia data on the Web. With the advanced version of the smartphone cameras, users are doing photography instantly. More than 1.2 trillion digital photos were taken worldwide in 2017. The embedded text within such images is in most cases multi-lingual and provides an important source of information for understanding their content. Detecting and recognizing text in such images are vital to a diverse set of applications ranging from assisting the blind and tourist guidance to autonomous-driving, Cyber security and visual search engines.

Besides the number of images, the multilingual nature of texts (language unknown in advance) poses another challenge of text detection. This text coexists everywhere in our daily lives. Like in product tags, street signs and guide information, many other environments contain more than two scripts text, what will be reflected in the captured images. Many applications need to access to this text regardless of its language. Even so script identification has been studied intensively and fairly high accuracy has been achieved, but it is not always free of errors. Moreover, the accuracy depends on the length of text, image type and quality. In born-digital and natural scene images, often multilingual texts co-occur in the same text line. And so, the text segment of one script/language is very short and the length of the text can not be considered as a script feature. Therefore we aim to exploit new geometric features and learning methods that can improve the identification accuracy for any type of text, but cannot guarantee error-free identification.

Many methods have followed a traditional approach for detecting the multi-script text. They build systems with separated blocks, starting by locating the text in the image and in a second stage identifying the script, eventually finding the language and recognizing the text from the cropped text images [76, 77, 78]. Text detection and script identification present two independent frameworks in these methods. Moreover, existing methods are optimized and fine-tuned at both preprocessing and feature extraction stages, and are usually database dependent. In order to overcome the above mentioned challenges, recent approaches for both tasks have considered automatic learning of features via different deep learning techniques. The penalty in these approaches is the fact that they build different networks for each task [78, 79]. Besides, they turned to holistic approach for the text detection task, and the whole image is fed to the network, which minimizes the loss of interesting regions and resulted in better performance results.

Robust and accurate text detection in weakly structured content is a challenging task in computer vision, particularly when dealing with multi-script texts. Analyzing the content in these images such as born-digital and scene images start by detecting the text. In this chapter, we propose a novel method for text detection and script identification in scene images and born-digital images. We opted for a holistic deep learning approach with a single network for both tasks, we believe that "the whole is greater than the sum of parts". Using the holistic approach permits the network to see the whole image and hence, to encode contextual information about the embedded text and its scripts. In our approach, textual feature extraction and script classification are performed jointly in one step. The relative features and their scripts are learned via a single convolutional neural network, the input data is the full image and the outputs are the text bounding boxes and the script of the detected text. Unlike traditional approaches for text detection, no pre-, post-processing steps are required for the input and output data. The CNN represents the main component in our method. It is inspired by YOLO: unified real-time object detection system [80]. YOLO frame object detection as a real-time regression problem to spatially separated bounding boxes and associated class probabilities. We follow its strategy and considered the text as the object to detect and its script as the final class.

The evaluation of the system is presented on different levels. Not many works consider the multi-lingual text, only few challenges are done for this specific task.

5.2 Prior Works of Text Detection Using YOLO System

Detecting the text in scene and born digital images is a challenging task due to two main problems. First, the high variations in text script and orientation, that leads to a complex layout with variable backgrounds and big variety of text (font, size, color). Secondly, we find the problems introduced by acquisition conditions like lighting, shadow and occlusion. Therefore, the traditional approaches based on hand-crafted features and regions filtering are not sufficiently robust to be generalized for multiple datasets and on different types of images.

Deep learning-based approaches, with features learned automatically, became the major trend in the recent years and very few methods followed a holistic approach as opposed to region-based ones. The text detection task can be posed either as an instance

segmentation task (output a mask of the text region) or as an object detection task (output coordinates of the text bounding box) as done in this work. While script identification approaches are classified into two main categories: structure-based or visual appearance-based. Two extensive reviews on text detection can be found in [81] and [82], and for script identification task in [83] and [84].

We note that very few works do joint text detection and script identification tasks [37] have . In the pipeline of an end-to-end system, the task of scene text detection is the first in the pipeline where regions of interest (ROI) are located based on features of the image. The text is then grouped at word or line level and only text image at this level will be saved. The text presents on a cropped image will then be fed to the script identification task.

Many works on text detection and recognition in images are inspired from the rapidly developing object detection algorithms. In [32], Liao et al. rely on an end-to-end trainable fully convolutional neural network to detect arbitrary-oriented text. The basic idea is inspired by the single-shot detector (SSD) object detection algorithm proposed in [85]. The main change consists of modifying the region proposal and bounding box regression modules of general detectors to localize text instances directly. While Bartz et al. [86] uses a spatial transformer networks (STN) to circularly attend the text at word level in the original image. In a second stage, the author proceeds to the recognition of the text at the cropped images level.

The biggest advantage of using YOLO: You Only Look Once Real-Time Object Detection system is it superb speed. It can process 45 frames per second. Moreover it can understand generalized object representation. That is why it is among the best algorithms for object detection and has shown similar performance to the R-CNN algorithms. View of this, many works are inspired from YOLO and use the same architecture for detecting text or other type of objects. From the examples of works on text detection we cite R-YOLO: A Real-Time Text Detector for Natural Scenes with Arbitrary Rotation [87]. First, it predicts the inclined bounding boxes and extract angle information for the oriented text boxes. Then based on the FCN structure of YOLOv4, it extracts the various features and determine the probability, confidence, and inclined bounding boxes of the text. Finally, it applies a modified version of the Non-Maximum Suppression technique to reduce the redundancy and generate the final output. Haifeng et al. [88] also use the same steps, but apply the YOLOv2 for learning the text features from the

natural scene images.

EAST [1] and YOLO [80] present two robust and rapid detector systems. Both concentrate efforts on designing loss functions and unified neural network for detecting and identifying the objects. Liu et al. [89] and Busta et al. [90] adopt their detection branches using CNN and CNN with LSTM respectively. For the recognition part, text proposals are fed into fixed tensors then transcribed into strings. Zhang et al. [91] propose a method composed of object-text detection network and text recognition network. YOLOv3 is used as the algorithm for the object-text detection task and CRNN is employed to deal with the text recognition task.

Bhunia et al. [92] present a script identification in natural scene image and video frames using an attention based Convolutional-LSTM. First, they extract global features using CNN-LSTM. Then, apply a patch-wise multiplication of these weights with corresponding CNN. To finally employ a fusion technique to weight the local and global features for an individual patch. Gomez et al. [93] also proposed a patch-based method for script identification in scene text images. The method utilized patch-based CNN features, and the Naive–Bayes Nearest Neighbor classifier.

Ashwaq et al. [94] use FCNs for both model enhancement and classification. They propose two end to end methods, namely: multi-channel mask (MCM) and multi-channel segmentation (MCS) inspired from EAST for joint text detection and script identification. The method presented good results mainly for large-sized and lengthy texts in scene images.

Our method proposes a unified CNN based on a holistic approach to prevent missing any possible text components. Therefore, discriminative features are learned from the original image via a CNN. Non-maximum suppression technique is used to aggregate the classified text boxes into words. Moreover, script identification task is performed simultaneously with the text detection task on the full image. Our method avoid making assumptions about the text orientation or script, hence, it could detect bilingual and/or multi-oriented text.

5.3 Joint Text Detection and Script Identification System

Script identification task is usually applied after the text detection. The output will be useful for the text recognition task, or presented as an independent task. Working on both tasks jointly need to adapt either the data or the architecture of the neural network (for example working with multi-channel networks). For this work doing joint text detection and script identification was considered as one task. And the proposed system can only process both tasks simultaneously and not one of them separately.

In this section, a detailed explanation of the proposed system is presented, followed by the experimental parameters and discussion of the results. Our method aims at detecting the text according to its script within a segmentation-free system. The main contribution of this work is to put in perspective the script identification within the text detection framework.

The system is mainly composed of one main component. It is based on a unique convolutional neural network able to learn powerful features from the images in the training phase. To train our model we proceed in a holistic approach and feed it with full images with embedded text from different scripts. The ground truth is like in most datasets, at word level. The final outputs are five coordinates presenting (x,y,w,h,s). Where (x,y,w,h) represents the coordinates of the text bounding box and s presents its script. The script is represented in the last fully connected layer as a number. For all the detected boxes in the image, we then apply a non-maximum suppression (NMS) to fix the multiple detections and generate the final output of the network.

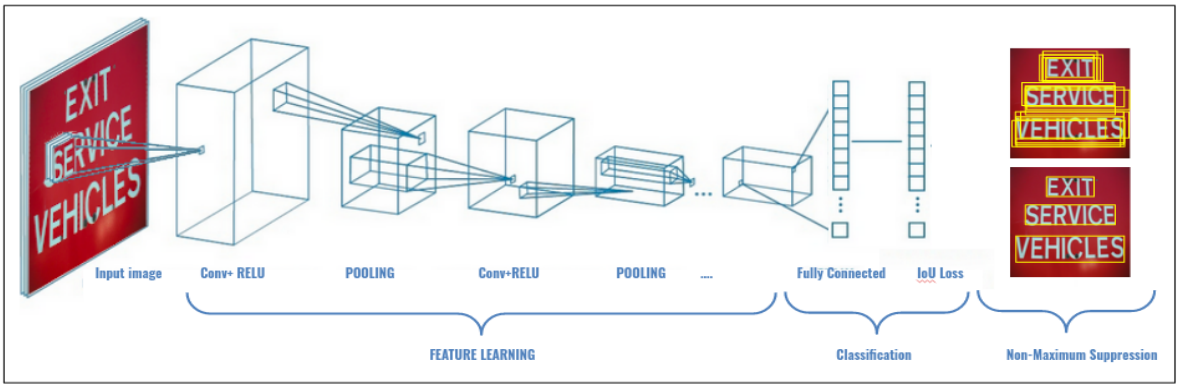

Figure 5.1: Block diagram of the proposed method with its two modules. First, Features learning with the CNN from SxS grids from the input image, followed by two fully connected layers. Then NMS to fix the multiple detections.

The block diagram of the method shows that the system has far fewer steps than

our first system presented in Section 4.3 and in other existing works. The CNN-based network extracts the features from the entire image to detect the script of the embedded text. It predicts each text bounding box across the different defined classes in the image simultaneously. This makes it robust and able to globally analyze the full image and all the textual objects in it. The script of the text is presented as the classification result. To determine the script of the text, we divide the image into equal cells and compute the confidence. If no object exists in that cell, the confidence scores are set to zero, otherwise, it is computed. The second step performs NMS on the multiple bounding boxes predicted per grid cell. This process generates the final output of the system which presents the bounding box of the detected text at word level.

In the next sub-sections, we present the CNN in detail with all the parameters and configurations. The method evaluation is then presented in Section 5.4.

5.3.1 Data Preparation

When working on the detection task whether for text or objects, the first step is to find the initial candidates. Deciding how to present relevant and meaningful candidates for the network remains very challenging mainly when working on text from different scripts. There are two approaches to prepare the initial data, either region-based approach or holistic approach. In Sub-section 4.3.1 we worked with region-based approach and tried to minimize the number of needed manipulations to get good regions. In this system, we choose holistic based approach that considers whole words as single units. This approach was introduced as an effective way to avoid segmentation errors.

The main challenge for the holistic approach is the computation complexity especially when dealing with important number of different words. For the training of our CNN based network described in the next section, we input the color images in their full size and the corresponding ground truth file. Each image is presented by one ground truth file containing the different bounding boxes of all the existing words. A bounding box is presented by its coordinates. The network handles any input image size of (w,h,3), where (w,h) presents the dimensions width and height and followed by the three color channels of the image.

Figure 5.2 illustrates the different manipulations applied to the image. The image is fed to the network in its real size then divided into SxS grids. These cells are then fed to

the different convolution layers to extract holistic features of the text regions. Then for each grid, we predict the bounding box BB of the eventual text object. For each grid cell we also compute its confidence score. it is presented as *Pr(Object)xIOU(Gtruth-pred)*. If no object exists in that cell, the confidence scores are set to zero. Otherwise, it is equal to the intersection over union (IOU) between the predicted bounding box and the ground truth. For each grid cell we then predict the possible bounding boxes that fit with. We predict the conditional class probabilities *Pr(Class|Object)*. These probabilities are then multiplied by the confidence score and a single box confidence is presented per cell. And with the computed probability, we decide if this box fit into a class (script) or not.

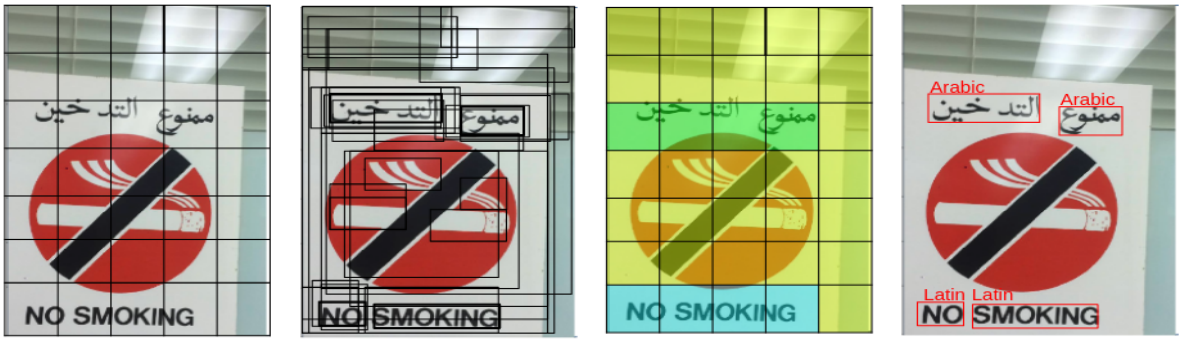

Figure 5.2: From left to right: The input image presented to the network in its full size is divided into SxS grid. Secondly, for each grid cell we predict the BB with the CNN. Third the probability for the box to be within a specific class. Last image presents the final output after NMS.

The data layer is also fed with the corresponding ground truth. In this approach, we defined the script of the text as its class. The number of classes is equal to the number of scripts we are working on. As an example, when working on bilingual text (Arabic/Latin), the number of classes is two: with (s = 0) for Arabic and (s = 1) for Latin. The same designation will be used in the ground-truth. Bounding boxes containing Arabic text will be prefixed by 0 and Latin ones prefixed by 1. The bounding box coordinates are also changed to the required format. In our case we provide the network with the coordinates of the center of the bounding box relative to the whole image.
Here we present the ground truth file of an image containing Arabic and Latin words. The original file presented below containing the coordinates of the text bounding boxes. The text files are space separated files, where each line corresponds to one word in the image and gives its bounding box coordinates and its transcription in the format:

left-top (x1,y1), right-bottom(x2,y2), script

Example of ground truth file used in the training phase:

158 128 411 181 Latin
443 128 501 169 Arabic
64 200 363 243 Arabic
394 199 487 239 Latin
72 271 382 312 Latin

The text files are space separated also, where each line corresponds to one word in the image and gives its script (s) and bounding box coordinates. The bounding box coordinates are presented by the coordinates of the center of the box and the relative width and height (width and height of the bounding box divided by the dimensions of the image) in the format:

s, x center, y center, relative w, relative h
1 0.444531 0.321875 0.395313 0.110417
0 0.7375 0.309375 0.090625 0.0854167
0 0.333594 0.461458 0.467187 0.0895833
1 0.688281 0.45625 0.145313 0.0833333
1 0.354687 0.607292 0.484375 0.0854167

Dataset balance has a significant impact on the performance of the CNN-based network. We detailed how we prepare the input data for the network. We only present bounding boxes with the classes to be detected. The network will then learn specific classes and all the rest will be considered as irrelevant regions.

5.3.2 CNN-based Model for Text Localization and Script Identification

At this stage, images are joined to their annotation files prepared in the previous section. For learning the text-script features, we used a convolution neural network. The advantage of the CNN compared to other neural networks is that it automatically detects the important features in the data. It is widely used for the object detection task as it can learn the key features for each class by itself, for example given many pictures for cats and dogs it can make the distinction easily. In this work, we define the text object by the script. As for the object detection problem, we consider the text is the object and the script is the class.

The CNN-based model is built based on the architecture of YOLO system (You Only Look Once). It is presented as a unified real-time object detection system proposed by Redmon et al [80]. The latter network was originally inspired by GoogLeNet network for image classification [95]. Both works focus on object detection and are extremely fast during testing time. The model process images in real-time at 45 frames per second. We built our model by adapting the architecture for learning textual features. Depending on the number of classes, many parameters have to be changed. We started by focusing on bilingual text to test the network and validate the full pipeline. We choose Arabic and Latin for our first experiments. Both scripts are among the most popular writing scripts and coexist in many countries. Joint text detection and script identification for Arabic-Latin text are useful for many applications. The final network proposed in Figure 5.3.

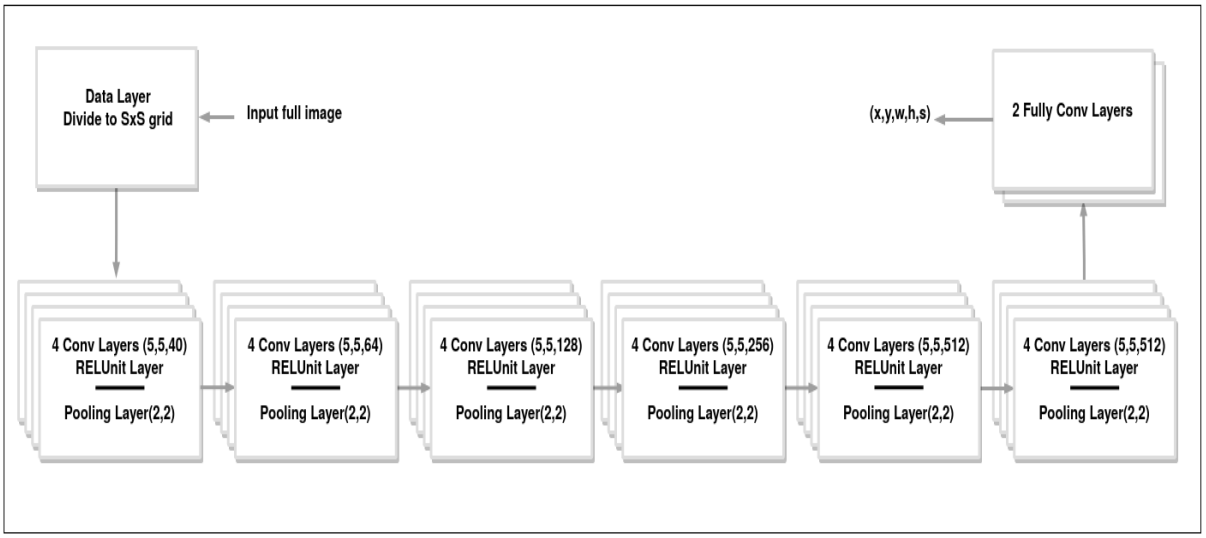

Figure 5.3: Structure of the CNN classifier network. The ConvUnit(w,h,n) represents a convolution layer of n features with wxh kernel size, connected to a ReLUnit layer and pooling layer with kernels of size 2x2. The structure ends with two fully connected layers.

The structure of the network is presented in Figure 5.3 is composed of a data layer followed by 24 convolutional 'Conv' layers. Each Conv layer is followed by a rectified linear unit layer 'RELUnit'. One pooling operation 2x2 pixels is applied after each block of 4 convolutions. Depending on the number of classes we compute the maximum number of batches and number of filters in the last convolutional layers. Then, the structure ends with 2 fully connected layers that generate the final output vector composed of the coor-

dinates of the detected bounding box and the class which represents the script of the text.

As for the feature learning, it goes through two phases. First, in the training phase, we fed the network with the training data and their corresponding labels as described in the previous section. The training data is presented as hundreds of images embedding multi-lingual text coupled with their modified ground truth annotation files. During the training process, different feature maps are generated from the Arabic and Latin text components at the different convolutional layers. By the end of the training phase, when the system reach the number of required iterations for convergence and after taking into consideration the confidence of all the boxes and the class probability, one bounding box with the highest class probability will be maintained. And the trained model is hence built to be used in the testing phase for classifying Latin text versus Arabic text.

The model is trained to detect all the text according to the defined scripts in the training phase. Overall, the network is able to handle both tasks simultaneously due to the way we present the data. Without any assumptions of the text type, orientation or script, the network learned the textual features related to each script. We presented the text data as a unique object regardless of its script. We assume that word text features from the same script have big similarities: adjacency between the letters, morphology of the letters within the same script, etc. Therefore, detecting the text according to its script resulted in good features able to make the difference between words from different scripts.

5.3.3 CNN-based Model Training and Output Optimization

At the end of the testing phase, we generate one file per image. It contains the coordinates of the detected text bounding boxes followed by their class identifier. The whole filtering process depends on this single threshold value. The selection of the threshold value is the key of the performance of the model. This threshold has been fixed experimentally. One cell in the image is classified to one unique class. However,due to the strategy followed of dividing the image into SxS grid cells, we obtain multiple boxes for a same word. The same cell can be detected multiple times in different combinations. Therefore, a filtering step for keeping the best bounding box per region is needed. First, only the bounding boxes with probability higher than the threshold are considered for further processing. Then to the resulting bounding boxes, we apply locality-aware Non-Maximum Suppression (NMS) to select the most appropriate bounding box predictions.

NMS is used in almost all state-of-the-art object detection pipelines.

In Figure 5.4, we display all the bounding boxes that have been detected for each word. The yellow boxes (one per word) presents the best boxes based on the detection and intersection over union scores.

Figure 5.4: The use of Non-Maximum Suppression to fix the multiple detection of the same zone of text. The first figure presents the output of the CNN. The second image presents the final detection result after the NMS step, where one box per word is chosen.

In this step, we present as input data the list of proposals detected boxes with their corresponding confidence scores and overlap threshold. NMS consists of greedily merging the higher scoring windows with lower scoring windows if they overlap enough. It starts by selecting the proposal with highest confidence score and add that box to the final list of filtered proposals. Then we compare this proposal with all the other proposals and compute the intersection over union IoU with each of them. Proposals with IoU greater than the fixed threshold will be considered in the list of final proposals. These steps will be repeated until there are no more proposals in the list of input.

5.4 Experiments and Performance Evaluation

We have implemented our method in a text detection and script identification system. Joining these two tasks with a single network permit to gain in processing time and consumed memory. However when we wanted to evaluate our work, there were no other methods working on the same challenge to compare with. Task 3 in Multi-Lingual scene

Text dataset publicly available in [3] is focusing on joint text detection and script identification. However, only simulations are done in this task and none of the described works are published. Therefore, we decided to evaluate our system on two levels.

First, we evaluate the system on joint text detection and script identification on the Multi-Lingual scene Text dataset publicly available in [3]. We start by testing on only Arabic-Latin scripts at first. Then, after making the required modifications for the training parameters, we tested on all the available scripts in the dataset. More details about the dataset and the implementation details are presented in the next sections.

Second, we present the results of the system for the text detection task. We make the evaluation on three public datasets of images from different categories. Two of them come from the ICDAR2015 Robust Reading Competition and the third dataset is ARAbic Scene Text Image 'ARASTI' (presented in Section 5.4.2). We made this choice in order to compare our work to state-of-the-art methods.

5.4.1 Implementation Details

When we start implementing this approach, we start by focusing on Arabic and Latin scripts only. Both scripts are very different. Considering the visual difference between writing Latin and Arabic scripts, we found the possibility to discriminate them based on their general structure and the number of occurrences among their structural characteristics. Indeed, in analyzing Arabic and Latin text, we can distinguish a difference in the cursivity and the number of diacritical dots in the Arabic script. But for printed Latin script, it is mainly composed of isolated letters. We concluded that features extracted from both scripts would be significant and specific for each script. So we tested on Arabic and Latin text only and set the parameters as detailed above.

All along the training and testing phases, the input images are fed to the proposed network in their original variable resolutions. To learn the features from this input we used the CNN presented in Figure 5.3 which shows the size of the feature maps and kernels of the different convolutional and pooling layers. The dense boxes from the CNN model are filtered based on the text confidence scores with the threshold value 0.8. Our network is trained by stochastic gradient descent with back-propagation and a maximum number of iterations of 10^6. The CNN solver parameters are as follows: Weight decay is $5\text{x}10^{-4}$ and momentum is 0.949. The loss is averaged each 100 iterations with a fixed learning rate during the training of $13\text{x}10^{-4}$.

In a second stage, we considered more scripts. We have developed the training model step by step. To build the training model, we started by Latin and Arabic scripts. Then

we added Chinese and Bangla scripts followed by Hindi and Japanese. Then the Korean and symbols classes. When we first trained the model on all the different scripts once, the texture features were not significant and powerful enough to discriminate the text from the different scripts. We train the model with the required data and save the model. That model will be used as checkpoint. These weights are then used to make predictions, or used as the basis for ongoing training.
For the configuring file of the training, it depends on the number of classes:

- $MaxBatches = (numberofclasses) \times 2000$
- $Steps = [(80\%ofMaxBatches), (90\%ofMaxBatches)]$
- $Filters = (numberofclasses \dot{+} 5) \times 3$

As for the evaluation of the system, for the text detection, we use the standard recall, precision and f-measure metrics proposed in the RRC competition [3] and used in most scene text detection works. A detected bounding box is considered as a match if it overlaps a ground truth bounding box by more than 50%. For the script identification task, we consider a correct case when the word is detected correctly and the script is also correct, and so the t_i which presents the detection result is similar to the ground truth g_i of the corresponding word [96]. The accuracy is computed as follows :

$$accuracy = \frac{1}{M} \sum_{i=1}^{M} \begin{cases} 1 & \text{if } g_i = t_i \\ 0 & \text{otherwise.} \end{cases}$$

where M is the total number of words in the testing dataset. We built the testing dataset with all the test images from the three datasets. So we got a mix of Arabic and Latin words at the same time. The experiments and results discussion are detailed in the following subsections.

For the experimental environment, we made the training and testing with:

- $Hardware\ environment$: Intel(R)Core(TM) I7-4790 3.60GHz CPU
- $Software\ environment$: Ubuntu 14.04 LTS
- cuda10.2, openCV2.4.20 and the framework Darknet

5.4.2 Database

Joint text detection and script identification task is tested on multi-lingual text. Tested and training images can include one or two scripts. Four public datasets are used for evaluating the performance of the method presented in this chapter. The goal from these different simulations is to prove two things.

First the efficiency of our system on detecting the text based on its script. For that we tested on multi-lingual text and verified the text detection accuracy and the script identification accuracy. Tests were done on the mixture of three datasets. Two of them contain Latin script and one is basically Arabic script.

Second, to prove the robustness of the system dealing with different types of images at the same time. Therefore, we tested on two different datasets of focused and incidental scene images and another dataset of born-digital images. These datasets present different categories of images with various properties. The embedded text is quite different in the scene images than in the born-digital images.

MLT dataset [96] : As described in chapter 3, the multi-lingual scene text detection and script identification dataset contain 20,000 real natural scene images with embedded text in seven scripts : Arabic, Latin, Chinese, Hindi, Japanese, Korean and Bangla. An eighth script class named "Symbols" was added for characters such as + / > :) ' . " - when they appear alone in a word. If a word combines two or more scripts, then it is considered as mixed. The images mainly contain intentional and focused scene text.

ARASTI dataset [97]: This dataset is collected from pictures of shopping areas, signboards, roadside signs, etc. It presents incidental scene text images that are very damaged. ARASTI dataset contains 374 images in total that we used for the testing phase. They contain a total of 1687 Arabic and Latin words. These images are similar to the first dataset on lighting, shadow and occlusion problems introduced by acquisition conditions.

Focused Scene Text and **Born-Digital Images** are the same used in the evaluation of the proposed method in Section 4.4.2. It is composed of 420 images, containing 3583 Latin words of more than 3 characters for the training set and 102 images, containing 918 words for the testing set.

5.4.3 Experimental Results

In order to show the efficiency of our method on the joint text detection and script identification tasks, we will present the detection results separately on each dataset and compare our work to state-of-the-art text detection methods.

For *Focused Scene Text* we present the results in Table 5.1. It shows the text detection results of our method applied on the RRC dataset [3]. Our method outperforms state-of-the-art methods by an F-score of 97.01%. We took the results from the Robust Reading Competition website. Figure 5.5 shows qualitative results of our method. The detected regions are in most cases precise and cover the whole word.

Table 5.1: Text detection results of the proposed method compared to state-of-the-art methods on the *Focused Scene Text* [3]

Method	Recall	Precision	F-measure
Our method	**95.10**	**98.9**	**97.01%**
[98]	94.67	96.44	95.54%
[99]	92.09	97.27	94.61%
[100]	82.28	89.94	85.94%

Figure 5.5: Examples of successful text detection results by the proposed method. Correctly detected text zones with the script from the Focused scene text dataset.

For the Born Digital images the results are presented in Table 5.2. Our method outperforms state-of-the-art method by an F-score of 96.81%. Figure 5.6 shows qualitative results of our method. The detected regions despite their very small and challenging size are detected with 96.61% of precision.

Table 5.2: Text detection results of the proposed method compared to state-of-the-art methods on the Born-Digital Images [3]

Method	Recall	Precision	F-measure
Our method	**97.01**	**96.61**	**96.81%**
[68]	91.00	95.00	93.00%
[69]	88.00	94.00	91.00%
[70]	88.38	91.87	90.09%

Figure 5.6: Examples of successful text detection and script identification results by the proposed method. Correctly detected text zones with the script from the Born-digital images.

For the ARASTI dataset, our method detected the text with 83.6% of F-measure, 90.71% of recall and 77.52% of precision. Figure 5.7 shows qualitative results of our method. The detected regions are precisely defined. However, some images are very damaged with blur and bad lighting. Most works on this dataset are for text recognition so no objective comparison with other methods of the literature using this dataset can

be done on the text detection task. The overall F-measure for our system is **92.47%** with a recall value of 95.64% and precision rate of 89.50%.

Figure 5.7: Examples of successful text detection and script identification results by the proposed method. Correctly detected text zones with the script from the ARASTI dataset.

For the script identification task, we evaluated our system using the accuracy formula cited in section 5.4.1. The final accuracy of our system is 93.77%. We tested the data once for both tasks, and only the detected text is evaluated for the script identification task. In order to compare our system for the joint text detection and script identification task for bi-lingual text, we tested our approach on the same dataset used in [94]. They used the Arabic and Latin text extracted from the MLT dataset [96]. We compare our proposed approach to the two methods presented in [94] : the multi-channel mask (MCM) and multi-channel segmentation (MCS) inspired from EAST." The results are presented in Table 5.3.

Then we tried a new experiment. Through all the steps of the system, we did not make any assumption of the text script or characteristics. We thought that applying the same system for the multi-lingual text (not only bi-lingual Arabic and Latin) is possible and we can achieve good results as for the bi-lingual text. Therefore, we started by preparing the MLT dataset for the experiment. On a first step, we changed the ground truth annotation from words bounding boxes presented by the 4 corners to their

Table 5.3: Joint text detection and script identification on the Arabic-Latin dataset.

Method	Dataset	Recall	Precision	F-measure
Our method		78.31	91.41	**84.35%**
MCS	MLT A-L	63.53	83.15	72.03%
MCM		63.44	82.29	71.64%
Our method	FST+BDI ARASTI	95.64	89.50	**92.47%**

presentation by the center of the bounding box coordinates and the relative dimensions, as described in Section 4.3.1.1. Then we do the training on the whole MLT dataset, and tested on the 7 different scripts. On a second step, we modified the parameters of the network according to the rules presented in Section 5.4.1. In Figure 5.8 we show some examples of successful results. And in Table 5.4 we presented our results and compared with two public works focusing on the joint text detection and script identification task for Baek et al. [101] and [98].

Figure 5.8: Examples of successful text detection and script identification results by the proposed method. Correctly detected text zones with the script from the MLT dataset.

Table 5.4: Joint text detection and script identification on the MLT dataset.

Method	Recall	Precision	Hmean
Our method	73.14	80.26	**76.53%**
Baek et al. [98]	60.50	78.52	68.34%
Baek et al. [101]	54.43	72.66	62.23%

5.5 Discussion

In this Chapter we presented a novel system for joint text detection and script identification in a simple segmentation-free holistic approach. Both tasks are done using a unified CNN-based features learning network. Our CNN generates a list of proposals bounding boxes with their corresponding confidence scores and overlap threshold. We applied non-maximum suppression technique to filter out the proposals and get the final output at word level.

Our idea is inspired from real-time object detection systems. Our goal is to detect all the text and identify the script to apply later the recognition task. Using a unified network, our system can be deployed for real-time applications as interactive tourists' guidance, where many scripts coexist simultaneously.

We have shown that adapting the object detection system YOLO to the problem of text detection and script identification can achieve good results. Moreover, our method is generalized to many types of images, while most works consider either focused scene or born-digital images since they have different characteristics. Furthermore, our system processes the full colour image as input and does not require tricky preprocessing steps. Overall, our proposed method has led to a powerful joint text detection and script identification system that performs better than state-of-the-art systems on different datasets.

Chapter 6

Conclusion and Perspectives

6.1 Summary of the PhD Thesis

This thesis has made a number of contributions with the objective of text detection and script identification using convolutional neural networks. Key contributions started with the building and the training of a convolutional neural network from scratch. Then we performed the fine-tuning of a model generated for object detection. We considered the text detection task as an object detection problem and the identification of the script jointly with the detection of the text on the full image. Finally, we created a multi-lingual text dataset. Let's discuss these points in more detail.

In Chapter 3 we present the multi-lingual text dataset. After a review of the existing datasets used for text analysis presented in Chapter 2, we decided to create a new dataset. This dataset presents an important contribution for the domain of text analysis in natural scene images and can be seen as the primary intellectual output of our research. The dataset has been cited in more than 300 papers and used by many more research works. The key advantage of this dataset besides its size and the diversity of scripts and languages it contains, is the data organisation that allows to execute many challenges. Four different challenges have been proposed during the *ICDAR 2019 Robust Reading Challenge on Multi-lingual scene text detection and recognition*: Text Localization, Script identification, Joint text detection and script identification and End-to-End text detection and recognition. New challenges will also be organised in the future.
Based on the performance of the existing works about text detection reviewed in Chapter 2 dedicated to the state-of-the-art and all along this manuscript, we conclude that using convolution neural networks with big datasets for training is effective for text de-

tection in natural scene images. Additionally the experiments show the importance of the quality and the manner we present the data to the network during the training process. It might have more impact on the final results than the depth of the network itself. Therefore, in our first system presented in Chapter 4 we focused our attention on presenting meaningful raw components to the network, and used a multi-level connected components extraction technique to achieve the text detection task. The multi-level CC analysis allows the extraction of redundant text and non-text components at multiple binarization levels to minimize the loss of any potential text candidates. The features of the resulting raw text/non-text components of different granularity levels are learned via a CNN. Then, the components classified as text at different granularity levels, are grouped in a graph based on the overlap of their extended bounding boxes. Then, the connected graph components are retained. When evaluated on different datasets our method achieved better detection results compared to state-of-the-art methods.

Based on the performance of our first system presented above, we start working on improving each step in the pipeline. Then we decided to work on a higher granularity level of the data, working at word level instead of connected component level. So we build the network in a way that it detects the text as an object defined by its script. Like in the child brain, reading involves more than simply deciphering words. It includes figuring out the relationship between the approximate 44 spoken sounds of the English language for example with the 26 letters of the alphabet, and over 150 spelling patterns. Knowing that, a child learns to read before writing or even recognizing the letters. However, a missed connection in the brain can lead to a complete misunderstanding of the text at hand and a very frustrated child who is learning to read. We build our system presented on Chapter 5 based on this idea. We tried to train our model the same way the child's brain learns. Therefore we presented the data to the network in the form of words, characterized only by their script. Each script has been converted into an object represented by thousands of words of that script. The convolutional neural network was able to generate meaningful features of each script, allowing the model to detect the script (implicitly the text) in the natural scene image. When evaluated on different datasets our method achieved good multi-lingual text detection results with correct identification of the script in most of the cases.

6.2 Perspectives and Future Works

There are many directions that could be explored in the work presented in this dissertation. In terms of algorithmic improvements, each method presented in this thesis can be improved. A possible improvement is to explore more elements in the image and to combine text understanding with object understanding. This can significantly improve the image annotations. Furthermore, we can respond to questions such as 'Can the text recognition improve object understanding?', and vice versa. This can be an interesting task to investigate in the future.

On the same context of our work presented in Chapter 5 focusing on joint text detection and script identification, we intend to upgrade our system to detect and recognise the text at the character level. In our actual system we perform the text identification and the script identification at word level. If we apply the same techniques at character level, we think that we will be able to perform the text detection, the text recognition and the identification of the script of the letter simultaneously.

We think that adding new scripts and languages in the Multi-Lingual Text dataset we be interesting to address new challenges and to improve the performances of methods developed for text detection and script identification. But a special focus on the "don't care" regions might be interesting. These regions in most of the cases contain damaged text: blurred, with low resolution, lighting variation, etc. The performance of all the existing methods are highly impacted by the regions of this category. Training a deep network on damaged texts but combined with a good ground truth annotation could generate significant features allowing the model to detect text in difficult cases. Moreover, there is a need to design a more robust evaluation protocols that can handle special appearances of text such as unfocused scene text, and also address sub-task evaluation for “don't care” words.

Bibliography

[1] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: an efficient and accurate scene text detector. In *IEEE conference on Computer Vision and Pattern Recognition*, pages 5551–5560, 2017.

[2] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i. Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazàn Almazàn, and Lluís Pere de las Heras. Icdar 2013 robust reading competition. In *ICDAR*, 2013.

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

[4] Hien Phuong Lai, Muriel Visani, Alain Boucher, and Jean-Marc Ogier. A new interactive semi-supervised clustering model for large image database indexing. *Pattern Recognition Letters*, 37:94–106, 2014.

[5] Hien Phuong Lai, Muriel Visani, Alain Boucher, and Jean-Marc Ogier. An experimental comparison of clustering methods for content-based indexing of large image databases. *Pattern Analysis and Applications*, 15(4):345–366, 2012.

[6] Mickael Coustaty, Vincent Courboulay, and Jean-Marc Ogier. Analyzing old documents using a complex approach: application to lettrines indexing. In *Advances in Knowledge Discovery and Management*, pages 155–171. Springer, 2012.

[7] Chee Kheng Ch'ng and Chee Seng Chan. Total-text: A comprehensive dataset for scene text detection and recognition. In *2017 14th IAPR International Conference*

on Document Analysis and Recognition (ICDAR), volume 1, pages 935–942. IEEE, 2017.

[8] Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu. Detecting texts of arbitrary orientations in natural images. In *2012 IEEE conference on computer vision and pattern recognition*, pages 1083–1090. IEEE, 2012.

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

[10] Seiichi Uchida. 25–text localization and recognition in images and video. *Handbook of Document Image Processing and Recognition*, pages 843–883, 2014.

[11] Qixiang Ye and David Doermann. Text detection and recognition in imagery: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 37(7):1480–1500, 2015.

[12] Xu-Cheng Yin, Ze-Yu Zuo, Shu Tian, and Cheng-Lin Liu. Text detection, tracking and recognition in video: a comprehensive survey. *IEEE Transactions on Image Processing*, 25(6):2752–2773, 2016.

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

[14] Trung Quy Phan, Palaiahnakote Shivakumara, and Chew Lim Tan. A laplacian method for video text detection. In *2009 10th International Conference on Document Analysis and Recognition*, pages 66–70. IEEE, 2009.

[15] Boris Epshtein, Eyal Ofek, and Yonatan Wexler. Detecting text in natural scenes with stroke width transform. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2963–2970. IEEE, 2010.

[16] Rong Huang, Palaiahnakote Shivakumara, and Seiichi Uchida. Scene character detection by an edge-ray filter. In *2013 12th International Conference on Document Analysis and Recognition*, pages 462–466. IEEE, 2013.

[17] Jung-Jin Lee, Pyoung-Hean Lee, Seong-Whan Lee, Alan Yuille, and Christof Koch. Adaboost for text detection in natural scene. In *2011 International conference on document analysis and recognition*, pages 429–434. IEEE, 2011.

[18] Xu-Cheng Yin, Wei-Yi Pei, Jun Zhang, and Hong-Wei Hao. Multi-orientation scene text detection with adaptive clustering. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1930–1937, 2015.

[19] Adam Coates, Blake Carpenter, Carl Case, Sanjeev Satheesh, Bipin Suresh, Tao Wang, David J Wu, and Andrew Y Ng. Text detection and character recognition in scene images with unsupervised feature learning. In *2011 International Conference on Document Analysis and Recognition*, pages 440–445. IEEE, 2011.

[20] Weilin Huang, Yu Qiao, and Xiaoou Tang. Robust scene text detection with convolution neural network induced mser trees. In *European conference on computer vision*, pages 497–511. Springer, 2014.

[21] Palaiahnakote Shivakumara, Rushi Padhuman Sreedhar, Trung Quy Phan, Shijian Lu, and Chew Lim Tan. Multioriented video scene text detection through bayesian classification and boundary growing. *IEEE Transactions on Circuits and systems for Video Technology*, 22(8):1227–1235, 2012.

[22] Zhenyu Zhao, Cong Fang, Zhouchen Lin, and Yi Wu. A robust hybrid method for text detection in natural scenes by learning-based partial differential equations. *Neurocomputing*, 168:23–34, 2015.

[23] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *International journal of computer vision*, 116(1):1–20, 2016.

[24] Chengquan Zhang, Cong Yao, Baoguang Shi, and Xiang Bai. Automatic discrimination of text and non-text natural images. In *ICDAR*, pages 886–890, 2015.

[25] Cong Yao, Xiang Bai, Nong Sang, Xinyu Zhou, Shuchang Zhou, and Zhimin Cao. Scene text detection via holistic, multi-channel prediction. *arXiv preprint arXiv:1606.09002*, 2016.

[26] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image with connectionist text proposal network. In *European conference on computer vision*, pages 56–72. Springer, 2016.

[27] Shi-Xue Zhang, Xiaobin Zhu, Jie-Bo Hou, Chang Liu, Chun Yang, Hongfa Wang, and Xu-Cheng Yin. Deep relational reasoning graph network for arbitrary shape text detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9699–9708, 2020.

[28] Dafang He, Xiao Yang, Chen Liang, Zihan Zhou, Alexander G Ororbi, Daniel Kifer, and C Lee Giles. Multi-scale fcn with cascaded instance aware segmentation for arbitrary oriented word spotting in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3519–3528, 2017.

[29] Wafa Khlif, Nibal Nayef, Jean-Christophe Burie, Jean-Marc Ogier, and Adel Alimi. Learning text component features via convolutional neural networks for scene text detection. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 79–84. IEEE, 2018.

[30] Wenhao He, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Deep direct regression for multi-oriented scene text detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 745–753, 2017.

[31] Yixin Li and Jinwen Ma. A unified deep neural network for scene text detection. In *International conference on intelligent computing*, pages 101–112. Springer, 2017.

[32] Minghui Liao, Baoguang Shi, and Xiang Bai. Textboxes++: A single-shot oriented scene text detector. *IEEE transactions on image processing*, 27(8):3676–3690, 2018.

[33] Minghui Liao, Zhen Zhu, Baoguang Shi, Gui-song Xia, and Xiang Bai. Rotation-sensitive regression for oriented scene text detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5909–5918, 2018.

[34] Yuchen Dai, Zheng Huang, Yuting Gao, Youxuan Xu, Kai Chen, Jie Guo, and Weidong Qiu. Fused text segmentation networks for multi-oriented scene text detection. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3604–3609. IEEE, 2018.

[35] Wenhai Wang, Enze Xie, Xiang Li, Wenbo Hou, Tong Lu, Gang Yu, and Shuai Shao. Shape robust text detection with progressive scale expansion network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9336–9345, 2019.

[36] Zhuotao Tian, Michelle Shu, Pengyuan Lyu, Ruiyu Li, Chao Zhou, Xiaoyong Shen, and Jiaya Jia. Learning shape-aware embedding for scene text detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4234–4243, 2019.

[37] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9365–9374, 2019.

[38] Kurban Ubul, Gulzira Tursun, Alimjan Aysa, Donato Impedovo, Giuseppe Pirlo, and Tuergen Yibulayin. Script identification of multi-script documents: A survey. *IEEE Access*, 5:6546–6559, 2017.

[39] Parul Sahare and Sanjay B Dhok. Script identification algorithms: a survey. *International Journal of Multimedia Information Retrieval*, 6(3):211–232, 2017.

[40] J. Gllavata and B. Freisleben. Script recognition in images with complex backgrounds. In *Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology, 2005.*, pages 589–594, 2005.

[41] Danni Zhao, Palaiahnakote Shivakumara, Shijian Lu, and Chew Lim Tan. New spatial-gradient-features for video script identification. In *2012 10th IAPR International Workshop on Document Analysis Systems*, pages 38–42. IEEE, 2012.

[42] Trung Quy Phan, Palaiahnakote Shivakumara, Zhang Ding, Shijian Lu, and Chew Lim Tan. Video script identification based on text lines. In *2011 International Conference on Document Analysis and Recognition*, pages 1240–1244. IEEE, 2011.

[43] Lluis Gomez and Dimosthenis Karatzas. A fine-grained approach to scene text script identification. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, pages 192–197. IEEE, 2016.

[44] Multi-language end-to-end. `https://github.com/lluisgomez/script_identification`. Accessed: 2021-12-30.

[45] Nabin Sharma, Sukalpa Chanda, Umapada Pal, and Michael Blumenstein. Word-wise script identification from video frames. In *2013 12th International Conference on Document Analysis and Recognition*, pages 867–871. IEEE, 2013.

[46] Linlin Li and Chew Lim Tan. Script identification of camera-based images. In *2008 19th International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008.

[47] Baoguang Shi, Xiang Bai, and Cong Yao. Script identification in the wild via discriminative convolutional neural network. *Pattern Recognition*, 52:448–458, 2016.

[48] Ikram Moalla. Caractérisation des écritures médiévales par des méthodes statistiques basées sur la cooccurrences. *University of Lyon, Thesis*, 2009.

[49] Baoguang Shi, Cong Yao, Chengquan Zhang, Xiaowei Guo, Feiyue Huang, and Xiang Bai. Automatic script identification in the wild. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 531–535. IEEE, 2015.

[50] Baoguang Shi, Xiang Bai, and Cong Yao. Script identification in the wild via discriminative convolutional neural network. *Pattern Recognition*, 52:448–458, 2016.

[51] Nabin Sharma, Ranju Mandal, Rabi Sharma, Umapada Pal, and Michael Blumenstein. Icdar2015 competition on video script identification (cvsi 2015). In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1196–1200. IEEE, 2015.

[52] Kaist scene text database. `http://www.iapr-tc11.org/mediawiki/index.php/KAIST_Scene_Text_Database`. Accessed: 2021-12-30.

[53] Ajeet Kumar Singh, Anand Mishra, Pranav Dabral, and CV Jawahar. A simple and effective solution for script identification in the wild. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, pages 428–433. IEEE, 2016.

[54] Deepak Kumar, MN Anil Prasad, and AG Ramakrishnan. Multi-script robust reading competition in icdar 2013. In *Proceedings of the 4th International Workshop on Multilingual OCR*, pages 1–5, 2013.

[55] Msra text detection 500 database. `http://www.iapr-tc11.org/mediawiki/index.php/MSRA_Text_Detection_500_Database_(MSRA-TD500)`. Accessed: 2021-12-30.

[56] Michal Bušta, Yash Patel, and Jiri Matas. E2e-mlt-an unconstrained end-to-end method for multi-language scene text. In *Asian Conference on Computer Vision*, pages 127–143. Springer, 2018.

[57] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2315–2324, 2016.

[58] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *2011 International Conference on Computer Vision*, pages 1457–1464. IEEE, 2011.

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

[60] Weilin Huang, Zhe Lin, Jianchao Yang, and Jue Wang. Text localization in natural images using stroke feature transform and text covariance descriptors. In *ICCV*, pages 1241–1248, 2013.

[61] Tao Wang, David J Wu, Adam Coates, and Andrew Y Ng. End-to-end text recognition with convolutional neural networks. In *ICPR*, pages 3304–3308, 2012.

[62] Siyu Zhu and Richard Zanibbi. A text detection system for natural scenes with convolutional feature learning and cascaded classification. In *CVPR*, pages 625–632, 2016.

[63] Fan Jiang, Zhihui Hao, and Xinran Liu. Deep scene text detection with connected component proposals. *arXiv preprint: 1708.05133*, 2017.

[64] M Meenakumari, T Mohanasundaram, R Suresh Kumar, A Maria Sindhuja, and S Gowdhamkumar. An efficient method for text detection and recognition in still images. *Annals of the Romanian Society for Cell Biology*, pages 7408–7415, 2021.

[65] Xiaobing Wang, Yonghong Song, and Yuanlin Zhang. Natural scene text detection with multi-channel connected component segmentation. In *12th International Conference on Document Analysis and Recognition*, pages 1375–1379, 2013.

[66] Kai Chen, Fei Yin, Amir Hussain, and Cheng-Lin Liu. Efficient text localization in born-digital images by local contrast-based segmentation. In *ICDAR*, pages 291–295, 2015.

[67] Caffe: Deep learning framework. `https://caffe.berkeleyvision.org/`.

[68] Yue Wu and Prem Natarajan. Self-organized text detection with minimal post-processing via border learning. In *International Conference on Computer Vision*, pages 5000–5009, 2017.

[69] Michael Villamizar, Olivier Canévet, and Jean-Marc Odobez. Multi-scale sequential network for semantic text segmentation and localization. *Pattern Recognition Letters*, 129:63–69, 2020.

[70] Nibal Nayef and Jean-Marc Ogier. Semantic text detection in born-digital images via fully convolutional networks. In *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 859–864. IEEE, 2017.

[71] Anna Zhu and Seiichi Uchida. Scene text relocation with guidance. In *ICDAR*, 2017.

[72] Zheng Zhang, Chengquan Zhang, Wei Shen, Cong Yao, Wenyu Liu, and Xiang Bai. Multi-oriented text detection with fully convolutional networks. *CoRR*, abs/1604.04018, 2016.

[73] Tong He, Weilin Huang, Yu Qiao, and Jian Yao. Text-attentional convolutional neural network for scene text detection. *IEEE transactions on image processing*, 25(6):2529–2541, 2016.

[74] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *IEEE Neural Information Processing Systems*, 2015.

[75] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems 29*, 2016.

[76] Ramanathan Rahul, Sreebha Bhaskaran, J Amudha, and Deepa Gupta. Multilingual text detection and identification from indian signage boards. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1120–1125. IEEE, 2018.

[77] HT Basavaraju, VN Manjunath Aradhya, and DS Guru. A novel arbitrary-oriented multilingual text detection in images/video. In *Information and decision sciences*, pages 519–529. Springer, 2018.

[78] Ankan Kumar Bhunia, Aishik Konwer, Ayan Bhunia, Abir Bhowmick, Partha P Roy, and Umapada Pal. Script identification in natural scene image and video frames using an attention based convolutional-lstm network. *Pattern Recognition*, 85:172–184, 2019.

[79] Gebrehiwot Kirubel. Multilingual text detection and script recognition from video scene using deep learning. *Addis Ababa University, Thesis*, 2019.

[80] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[81] Nishant Subramani, Alexandre Matton, Malcolm Greaves, and Adrian Lam. A survey of deep learning approaches for ocr and document understanding. *arXiv preprint arXiv:2011.13534*, 2020.

[82] Qixiang Ye and David Doermann. Text detection and recognition in imagery: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 37(7):1480–1500, 2014.

[83] Miral V Donda, Harshadkumar B Prajapati, and Vipul K Dabhi. Survey on automatic script identification techniques. In *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, pages 1–5. IEEE, 2019.

[84] Parul Sahare and Sanjay B Dhok. Script identification algorithms: a survey. *International Journal of Multimedia Information Retrieval*, 6(3):211–232, 2017.

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

[86] Christian Bartz, Haojin Yang, and Christoph Meinel. See: towards semi-supervised end-to-end scene text recognition. In *Thirty-second aaai conference on artificial intelligence*, 2018.

[87] Xiqi Wang, Shunyi Zheng, Ce Zhang, Rui Li, and Li Gui. R-yolo: A real-time text detector for natural scenes with arbitrary rotation. *Sensors*, 21(3):888, 2021.

[88] Dong Haifeng and Han Siqi. Natural scene text detection based on yolo v2 network model. In *Journal of Physics: Conference Series*, volume 1634, page 012013. IOP Publishing, 2020.

[89] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. Fots: Fast oriented text spotting with a unified network. In *IEEE conference on computer vision and pattern recognition*, pages 5676–5685, 2018.

[90] Michal Busta, Lukas Neumann, and Jiri Matas. Deep textspotter: An end-to-end trainable scene text localization and recognition framework. In *IEEE international conference on computer vision*, pages 2204–2212, 2017.

[91] Fan Zhang, Jiaxing Luan, Zhichao Xu, and Wei Chen. Detreco: Object-text detection and recognition based on deep neural network. *Mathematical Problems in Engineering*, 2020, 2020.

[92] Ankan Bhunia, Aishik Konwer, Ayan Bhunia, Abir Bhowmick, Partha P Roy, and Umapada Pal. Script identification in natural scene image and video frames using an attention based convolutional-lstm network. *Pattern Recognition*, 85:172–184, 2019.

[93] Lluis Gomez, Anguelos Nicolaou, and Dimosthenis Karatzas. Improving patch-based scene text script identification with ensembles of conjoined networks. *Pattern Recognition*, 67:85–96, 2017.

[94] Ashwaq Khalil, Moath Jarrah, Mahmoud Al-Ayyoub, and Yaser Jararweh. Text detection and script identification in natural scene images using deep learning. *Computers & Electrical Engineering*, 91:107043, 2021.

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

[96] Nibal Nayef, Yash Patel, Michal Busta, Pinaki Nath Chowdhury, Dimosthenis Karatzas, Wafa Khlif, Jiri Matas, Umapada Pal, Jean-Christophe Burie, Cheng-lin Liu, et al. Icdar2019 robust reading challenge on multi-lingual scene text detection and recognition—rrc-mlt-2019. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1582–1587. IEEE, 2019.

[97] Maroua Tounsi, Ikram Moalla, and Adel M Alimi. Arasti: A database for arabic scene text recognition. In *2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*, pages 140–144. IEEE, 2017.

[98] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9365–9374, 2019.

[99] Ye Jian, Chen Zhe, Liu Juhua, and Du Bo. Textfusenet: Scene text detection with richer fused features. In *IJCAI*, pages 516–522, 2020.

[100] Khlif Wafa, Nayef Nibal, Burie Jean-Christophe, Ogier Jean-Marc, and Alimi Adel. Learning text component features via convolutional neural networks for scene text detection. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 79–84. IEEE, 2018.

[101] Jeonghun Baek, Geewook Kim, Junyeop Leeand, Sungrae Park Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. What is wrong with scene text recognition model comparisons? dataset and model analysis. *CoRR*, abs/1904.01906, 2019.