UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Deep Learning-Based Text Detection and Recognition

by Qingqing Wang

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of Xiangjian He & Michael Blumenstein & Wenjing Jia

University of Technology Sydney
Faculty of Engineering and Information Technology

April, 2020

Certificate of Authorship/Originality

I, Qingqing Wang, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy in the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise reference or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of the requirements for a degree at any other academic institution except as fully acknowledged within the text. This thesis is the result of a Collaborative Doctoral Research Degree program with East China Normal University.

This research is supported by the Australian Government Research Training Program.

Production Note:

Signature: Signature removed prior to publication.

Date: 7/4/2020

Acknowledgements

I would like to acknowledge all people and institutions that have provided support during my PhD candidature.

First of all, I thank my supervisors Prof. Yue Lu, Prof. Xiangjian He and Prof. Michael Blumenstein for their patient advice and guidance in my academic career. They are all knowledgeable, enthusiastic, optimistic, wise and helpful people. I have learned so much from them, not only how to be a successful scholar but also how to be a kind people in life. My supervisors have provided me such a precious opportunity to study overseas at the University of Technology Sydney (UTS). I never thought that such a cool thing would happen to me. Prof. Prof. Yue Lu is one of the most important people in my life. I jointed his research team in 2013 as a newbie in research. Since then, Prof. Yue Lu has taught me how to do research from scratch and led me to the right academic way. I know Prof. Yue Lu has always been doing his best for me and believe his knowledge, experience and vision enable him to help me do the best choice in every crossroad of my life. I am honoured to be supervised by Prof. Xiangjian He and Prof. Michael Blumenstein in UTS. Both of them are nice and humble in life and with very good reputation in our research community. Studying abroad alone is not easy, but with their help and support, I have enjoyed a wonderful time in Sydney.

I would also like to express gratitude to Dr. Wenjing Jia for her valuable comments on my writing skill. I believe what I have learned from her will benefit me for life. Additional thanks to students in my LPAIS lab and CVPR lab, including Wenqiao Sun, Hongjian Zhan, Xuecheng He, Yuhuang Xiu, Xiaohua Wei, Ye Huang, Xiaochen Fan, Yue Xi, Lei Liu, Saeed Amirgholipour Kasmani, Hesam Hesamian, etc. They have altogether created a wonderful research environment and made me enjoy my research in East China Normal University (ECNU) and UTS. The

days of pursuing a dual PhD degree are sometimes full of darkness, loneliness and depression. Thank my friends Qin Chen, Ye Huang and others for accompanying and supporting me during these tough times.

Next, I am grateful for the financial supports from ECNU, UTS, Chinese Scholarship Council (CSC) and BYKER DIGITAL BIOTECHNOLOGY CO., LTD. They make me relaxed on my way to dream.

Finally, I sincerely thank my parents and sisters for their love, understanding and encouragement in this long journey. Their support is always my strength and lets me feel free to pursuit my dream. Being a member of such a wonderful family is the most beautiful thing that happened to me. I am so proud of you all and thank you very much for everything you have done for me.

Qingqing Wang
Sydney, Australia, 2020.

List of Publications

Published Papers

1. **Qingqing Wang**, Ye Huang, Wenjing Jia, Xiangjian He, Michael Blumenstein, Shujing Lyu and Yue Lu. FACLSTM: ConvLSTM with focused attention for scene text recognition. SCIENCE CHINA Information Science, 2020, 63(2): 120103. (JCR: Q1)

2. **Qingqing Wang**, Wenjing Jia, Xiangjian He, Yue Lu, Michael Blumenstein, Ye Huang and Shujing Lyu. DeepText: detecting text from the wild with multi-ASPP-assembled DeepLab. *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 208-213, 2019. (Top: A)

3. **Qingqing Wang**, Wenjing Jia, Xiangjian He, Yue Lu, Michael Blumenstein, Ye Huang and Shujing Lyu. ReELFA: a scene text recognizer with encoded location and focused attention. *International Conference on Document Analysis and Recognition Workshops(ICDARW)*, pp. 71-76, 2019. (Top: A)

4. Yuhuan Xiu, **Qingqing Wang**, Hongjian Zhan, Yue Lu and Man Lan. A Handwritten Chinese Text Recognizer Applying Multi-level Multimodal Fusion Network. *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1464-1469, 2019. (Top: A)

5. **Qingqing Wang** and Yue Lu. A sequence labeling convolutional network and its application on handwritten string recognition. *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2950-2956, 2017. (Top: A*)

6. **Qingqing Wang** and Yue Lu. Similar handwritten Chinese character recognition using hierarchical CNN model. *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 603-608, 2017. (Top A)

7. **Qingqing Wang**, Xiao Tu, Shujing Lu and Yue Lu. Text extraction from mail images with complex background. *International Forum of Digital TV and Wireless Multimedia Communication (IFTC)*, pp. 3-11, 2017.

8. Hongjian Zhan, **Qingqing Wang** and Yue Lu. Handwritten digit string recognition by combination of ResNet and RNN-CTC. *International Conference on Neural Information Processing (ICONIP)*, pp. 583-591, 2017. (Top A)

9. Ruyu Zhang, **Qingqing Wang** and Yue Lu. Combination of ResNet and center loss based metric learning for handwritten Chinese character recognition. *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 25-29, 2017. (Top A)

10. Shangxuan Tian, Ujjwal Bhattacharya, Shijian Lu, Bolan Su, **Qingqing Wang**, Xiaohua Wei, Yue Lu and Chew Lim Tan. Multilingual scene character recognition with co-occurrence of histogram of oriented gradients. Pattern Recognition, 51(2016), pp. 125-134. (JCR: Q1)

11. **Qingqing Wang**, Yue Lu and Ying Wen. Scene text detection using sequential nontext filtering. *International Conference on Image Processing (ICIP)*, pp. 1742-1746, 2015. (Top A)

12. **Qingqing Wang**, Yue Lu and Shiliang Sun. Text detection in nature scene images using two-stage nontext filtering. *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 106-110, 2015. (Top A)

Contents

List of Figures

List of Tables

Abbreviation

CC - Connected Component

MSER - Maximal Stable Extremal Region

LSTM - Long Short-Term Memory

CNN - Convolutional Neural Network

FC-LSTM - Fully-connected-LSTM

SPP - Spatial Pyramid Pooling

ASPP - Atrous Spatial Pyramid Pooling

IoU - Intersection-over-Union

BN - Batch Normalization

RNN - Recurrent Neural Network

ReELFA - Recognizer with Encoded Location and Focused Attention

ConvLSTM - Convolution LSTM

FACLSTM - ConvLSTM with Focused Attention

HMM - Hidden Markov Model

ICDAR - International Conference on Document Analysis and Recognition

HOG - Histogram of Oriented Gradients

LBP - Local Binary Pattern

SWT - Stroke Width Transform

ERs - Extremal Regions

SVM - Support Vector Machine

RF - Random Forests

FCN - Fully Convolution Network

SSD - Single Shot Multibox Detector

RPN - Region Proposal Network

NMS - Non-maximum Suppression

RoI - Region of Interest

FPN - Feature Pyramid Networks

TCM - Text Context Module

RRPN - Rotated Region Proposal Network

CTPN - Connectionist Text Proposal Network

ITN - Instance Transformation Network

CTC - Connectionist Temporal Classification

STN - Spatial Transformer Network

EP - Edit Probability

HAM - Hierarchical Attention Mechanism

TPS - Thin-Plate-Spline

HTR - Handwritten Text Recognition

Semi-CRFs - Semi-Markov Conditional Random Fields

FNNLM - Feedforward Neural Network Language Model

RNNLM - Recurrent Neural Network Language Model

MDLSTM - Multi-Dimensional Long-Short Term Memory

PCA - Principal Components Analysis

WFST - Weighted Finite-State Transducer

HCR - Handwritten Character Recognition

MQDF - Modified Quadratic Discriminant Function

ATRCNN - Alternately Trained Relaxation Convolutional Neural Network

DirectMap - Direction-decomposed Feature Map

AP - Average Precision

MST - Minimum Spanning Trees

CRF - Conditional Random Field

SVHN - Street View House Number

NAS - Neural Network Search

PAN - Pyramid Attention Network

PMTD - Pyramid Mask Text Detector

ABSTRACT

Deep Learning-Based Text Detection and Recognition

by

Qingqing Wang

Texts play a critical role in our daily life. They are everywhere such as slogans on posters, licence plates on cars, etc., to transmit information and knowledge. With the popularity of mobile devices with cameras, more and more texts are collected, transmitted and stored as text images. Automatically reading texts from images is of high application potentials. Therefore, related researches have been attracting considerable attentions from the computer vision community. Scene texts and handwritten texts are the two most difficult texts to be automatically read because of the challenges posed by the complexity of backgrounds, the uncertainty of capturing conditions, the diversity of text appearances, touching characters and the variety of handwriting styles.

Text detection, *i.e.,* localizing text areas from images, and text recognition, *i.e.,* transcribing located text areas into character sequences, are two key steps of robust text reading. In recent years, they have entered a deep learning era, where Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) play important roles. Here, we conduct researches on text detection and recognition based on CNN and LSTM, as presented below.

1. To improve the recall rate of small text areas in oriented text detection, we propose an Xception-based multi-ASPP-assembled scene text detector named DeepText. DeepText inserts multiple Atrous Spatial Pyramid Pooling (ASPP) modules into Xception after feature maps with different resolutions to retain richer information for small text areas, and introduces auxiliary connections and auxiliary losses to speed up convergence and boost the dis-

crimination ability of lower encoder layers.

2. To address the issue that Mask R-CNN cannot fully leverage global information when performing predictions, we propose a scene text detector named GMask R-CNN, where a global mask module is designed to perform semantic segmentation by considering global information.

3. To tackle the problem that LSTM neglects the valuable spatial and structural information of 2-D text images, we propose two scene text recognisers named FACLSTM, which exploits convolution LSTM to directly perform sequential transcription in 2-D space, and ReELFA, which utilizes one-hot encoded locations to enhance features with pixels' spatial information.

4. To solve the problem that CNNs with fully connected layers are not suitable for sequential prediction tasks due to their requirements of fixed-size inputs/outputs, we propose a CNN-based handwritten text recogniser CFSPP. CFSPP embeds a Spatial Pyramid Pooling-based intermediate layer between convolutional layers and fully connected layers to convert arbitrary-size feature maps into feature vectors with specific lengths.

Chapter 1

Introduction

In this chapter, we give a general knowledge of text detection and recognition as well as a brief introduction of this thesis, including research backgrounds, research motivations, related works, widely used evaluation metrics, contributions of our work, thesis organisations, etc.

1.1 Background of Text Detection and Recognition

Text is one of the most brilliant inventions in the history of human civilization and has been playing a critical role in recording information, conveying knowledge and facilitating communication. With the dramatic development of techniques, especially ones on mobile devices, images have become essential carriers of texts. Therefore, automatically reading texts from images is of high application potentials, such as automatic navigation, intelligent transportation system, blind assistance system, image retrieval, digitization of historical documents, machine translation for texts present in images, etc. On the other hand, as shown in an experimental study [51], humans are more likely to fixate on texts when they view images containing both texts and other objects, which further improves the importance of texts. Thus, researches on automatically reading texts from images will undoubtedly benefit humans' well-being.

According to literature, text detection, *i.e.,* localizing text from images, and text recognition, *i.e.,* transcribing text images into human readable ASCII characters, are the two primary focuses of current researches on robust text reading in the field of computer vision. The basis of performing text detection and text recognition is the structural information of text strokes. As we all know, texts are constructed by a series of strokes and strokes are connected components (CCs) with pairs

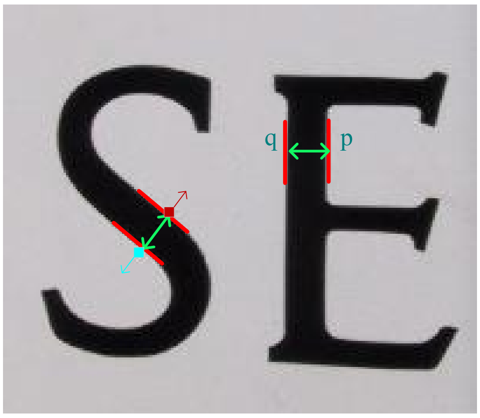
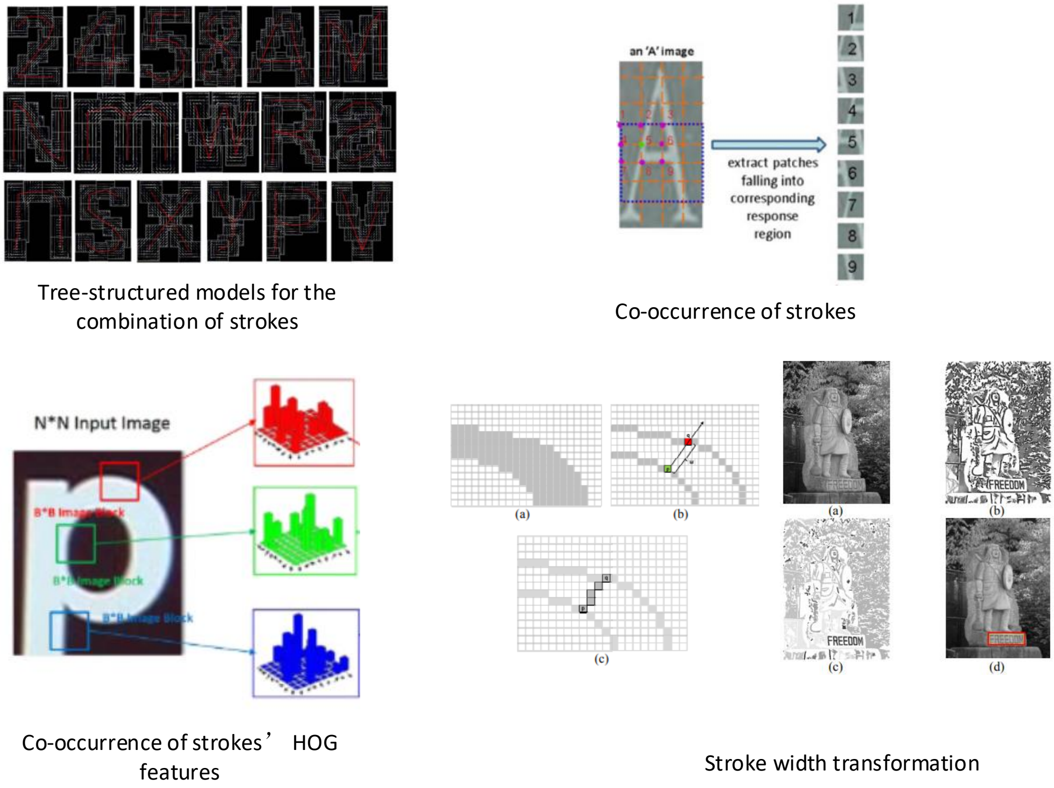


Figure 1.1 : Texts are the combination of strokes.


Tree-structured models for the combination of strokes

Co-occurrence of strokes

Co-occurrence of strokes' HOG features

Stroke width transformation

Figure 1.2 : Stroke-related features used in [111, 30, 120, 27] for text detection and recognition.

of opposite edges, like edge p and edge q on the letter 'E' in Fig. 1.1. Besides, widths of strokes, *i.e.,* distances between opposite edges, belonging to the same character or word usually keep consistent. Therefore, in literature, stroke-related features including co-occurrence of strokes, histogram of oriented gradients (HOG) of strokes, aspect ratio of the combined strokes, consistency of stroke width and intensity, etc., are the most commonly used features for both text detection and text recognition, as shown in Fig. 1.2.

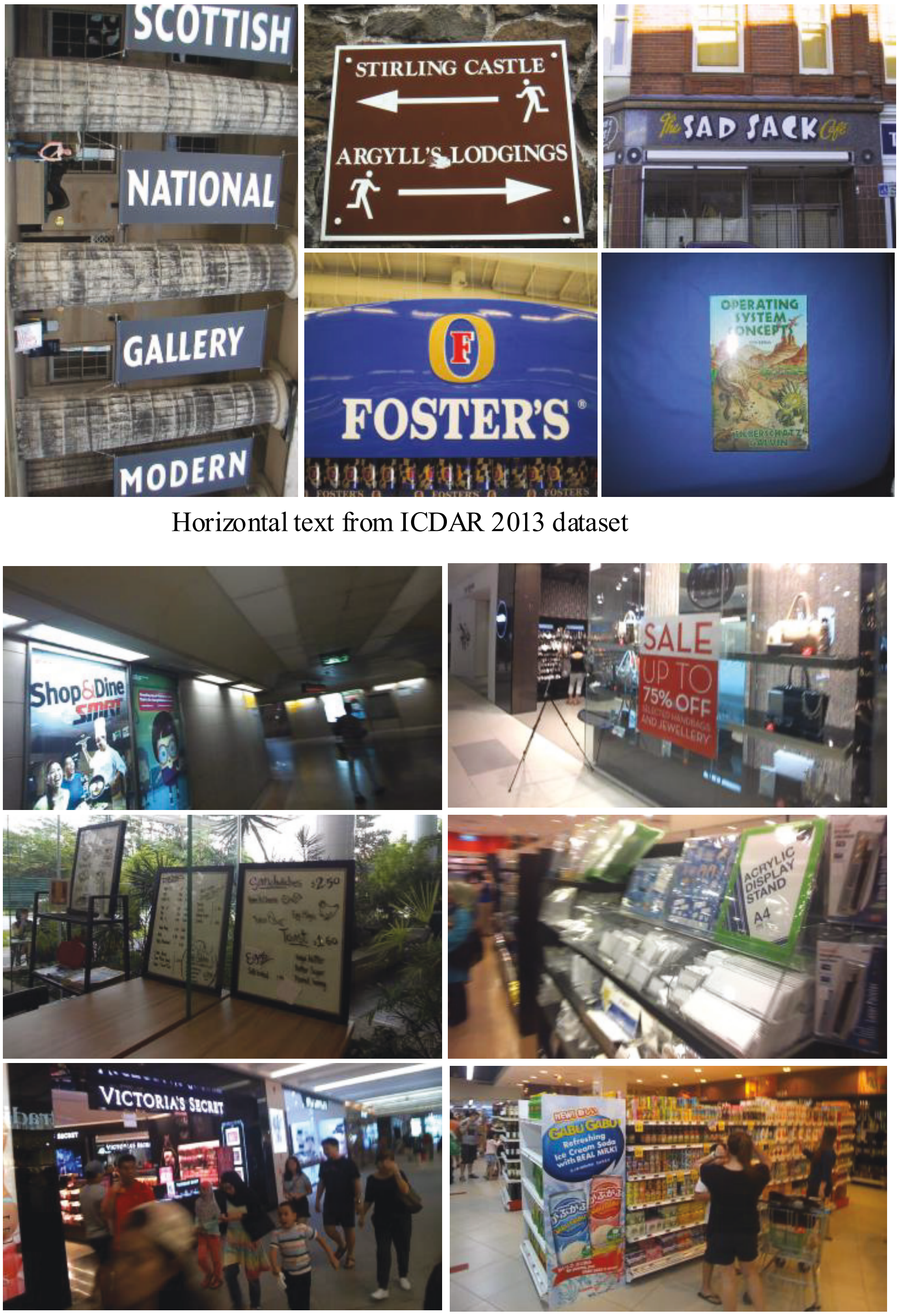
In our daily life, texts mainly present in two forms, *i.e.,* scene texts and handwritten texts. Scene texts refer to texts appearing in scene images, which are captured in the wild without constraints, such as images shown in Fig. 1.3 and Fig. 1.4. Challenges of handling scene texts mainly arise from the three aspects listed as follows.

- Complicated backgrounds. Texts could present anywhere together with any scene contexts, such as houses, bags, cars, bottles, trees, animals, etc., among which many may have similar texture as texts like bricks and fences. False positives caused by the complicated backgrounds severely affect models' performance of discriminating texts from other objects.

- Unrestricted capturing conditions, including uneven illumination, perspective distortion, low resolution, motion blur, skew, etc. These factors make texts harder to be detected and recognized.

- Various text appearances in terms of colours, sizes, fonts, directions, scales, shapes and so forth. In other words, scene texts suffer from huge intra-class variance, which introduces extra challenges to models' representation abilities and robustness.

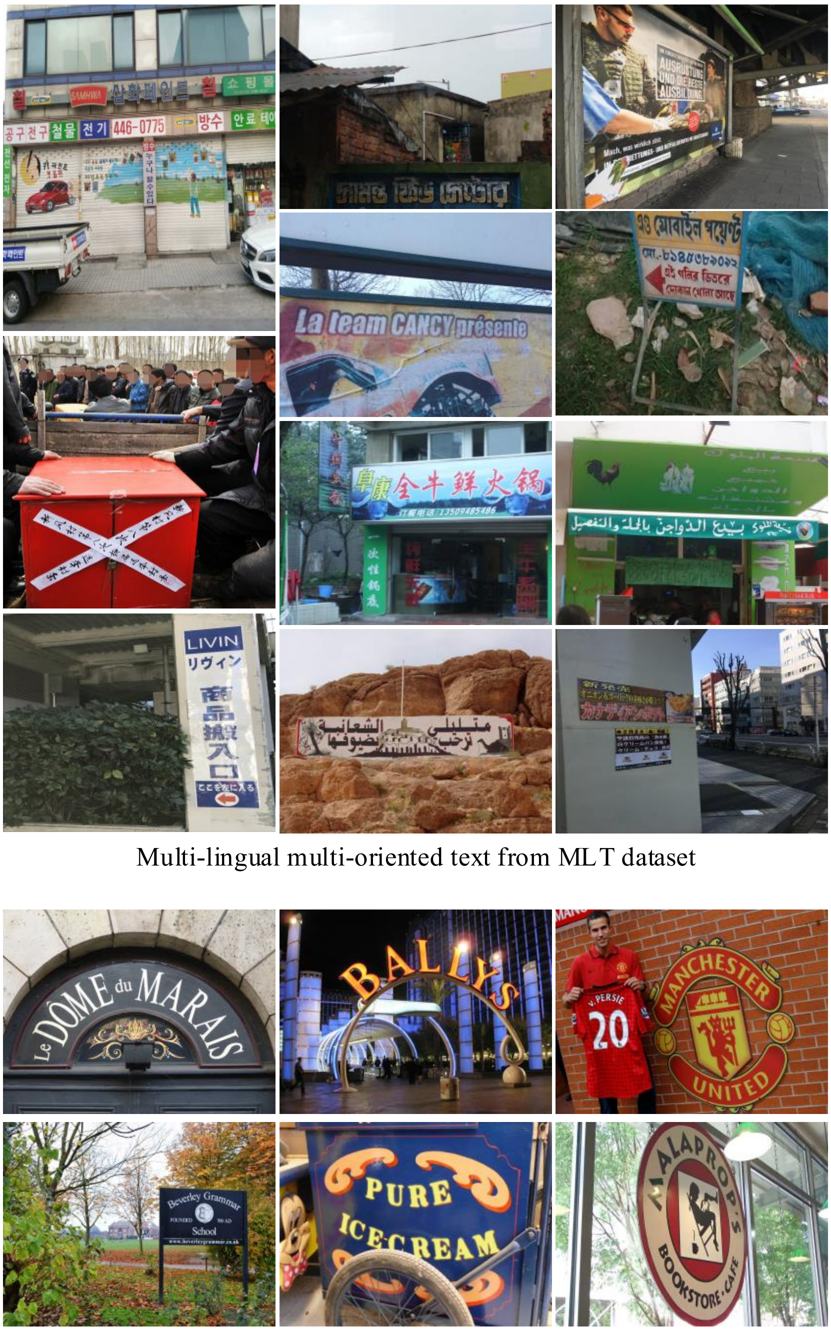
On the other hand, handwritten texts are texts written by human beings on papers or touch screens, such as signatures and money amounts on bank cheques, phone numbers and mail addresses on express waybills, notes on papers, comments on historical documents, etc. As shown in Fig. 1.5, handwritten texts are usually



Horizontal text from ICDAR 2013 dataset

Oriented text from ICDAR 2015 dataset

Figure 1.3 : Scene images with horizontal text (selected from ICDAR 2013 dataset [53]) and oriented text (selected from ICDAR 2015 dataset [52]).



Multi-lingual multi-oriented text from MLT dataset

Curved text from Total-text dataset

Figure 1.4 : Scene images with multi-lingual multi-oriented text (selected from MLT dataset [91]) and curved text (selected from Total-text dataset [18]).

with uniform appearances and lie over relatively clean backgrounds, and their picturing conditions are usually under control. However, even so, reading handwritten texts is still full of challenges because of their following characteristics.

- Diverse writing styles. Everyone has his unique writing style, which is the basis of writer identification. However, for handwritten text recognition, diverse writing styles represent huge intra-class variance and higher requirement on model's robustness and discriminative abilities.
- Touching characters. Strokes of adjacent characters often connect to each other because of human writing habits, and thus shapes of these characters may be significantly changed. To deal with these touching characters, special and complicated operations are sometimes required because it is almost impossible to correctly separate them.

In our research community, related benchmark datasets are usually collected according to texts' languages and orientations. For example, ICDAR 2013 [53] dataset and ICDAR 2015 [52] dataset are English horizontal scene text dataset and English oriented scene text dataset, respectively, while ICDAR 2017 [91] dataset, *i.e.,* MLT, is a multi-lingual multi-oriented dataset, where 9 languages are involved, including English, Chinese, Japanese, Korean, French, Arabic, Italian, German and Indian. In Table 1.1, we give a brief description to benchmark scene text datasets that widely used in last decade (according to [81]) and related samples are shown in Fig. 1.3 and Fig. 1.4. Note that in literature, detection results of horizontal texts, oriented texts and curved texts are represented by horizontal rectangles, convex quadrangles (or rectangles with angels) and polygons, respectively.

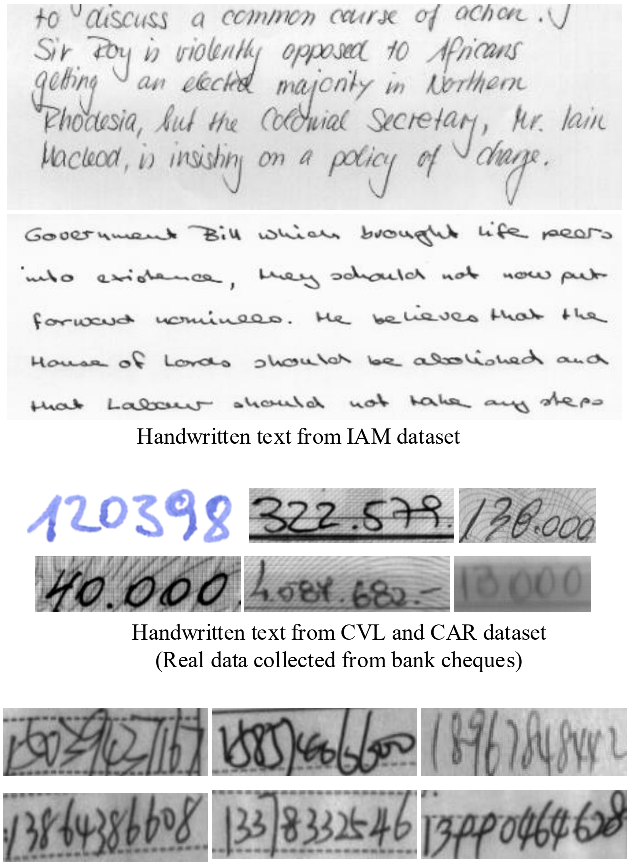
Since handwritten texts usually lie over relatively clean backgrounds, and are not troubled by the detection problem, the existing handwritten text datasets are mainly proposed for the recognition task. In Table 1.2, we list some widely used datasets that published in last decade for ICDAR handwritten recognition competitions, together with their recognition performances reported in these competitions. Corresponding samples are presented in Fig. 1.5.

Table 1.1 : Scene text datasets that widely used in last decade. EN and CN denote English and Chinese, respectively.

Dataset (Year)	Image Num (train/test)	Text Num (train/test)	orientations	Language	Detection	Recognition
SVT(2010)	100/250	257/647	Horizontal	EN	✓	✓
SVHN(2010)	73257/26032	73257/26032	Horizontal	Digits	-	✓
MSRA-TD500(2012)	300/200	1068/651	Oriented	EN, CN	✓	-
IIIT5K(2012)	2000/3000	2000/3000	Horizontal	EN	-	✓
SVTP(2013)	-/639	-/639	Oriented	EN	-	✓
ICDAR 2013(2013)	229/233	848/1095	Horizontal	EN	✓	✓
CUTE(2014)	-/80	-/-	Curved	EN	✓	✓
ICDAR 2015(2015)	1000/500	-/-	Oriented	EN	✓	✓
ICDAR RCTW(2017)	8034/4229	-/-	Oriented	CN	✓	✓
Total-Text(2017)	1255/300	-/-	Curved	EN, CN	✓	✓
MLT(2017)	9000/9000	-/-	Oriented	9 languages	✓	-
CTW1500(2017)	1000/500	-/-	Curved	EN	✓	-

Table 1.2 : Handwritten text datasets that published in last decade and their recognition performance in ICDAR competitions. Evaluation metrics are word-level accuracy except ones marked by CRA (character level accuracy).

Dataset(Year)	Language	Word Num/String Num/Vocabulary Size	Accuracy
CASIA-OLHWDB(2011) [69]	Chinese	224419/91563/-	70.63
CASIA-OLHWDB(2013) [146]	Chinese	224419/91563/-	89.28
ORAND-CAR & CVL(2014) [26]	Digits	-/19689/-	78.72
TRANSCRIPTORIUM(2015) [103]	English	186643/21752/1067	69.8
READ(2016) [101]	German	43460/10550/8120	79.1
READ(2017) [102]	German	1798535/209146/108813	82.9
READ(2018) [114]	German, Italian	98239/16984/-	82.1 (CRA)


to discuss a common course of action.
Sir Roy is violently opposed to Africans
getting an elected majority in Northern
Rhodesia, but the Colonial Secretary, Mr. Iain
Macleod, is insisting on a policy of change.
Government Bill which brought life peers
into existence, they should not now put
forward nominees. He believes that the
House of Lords should be abolished and
that Labour should not take any steps
120398
322.579
128.000
40.000
4.084.680.-
13000
13864386608
13378332546

Handwritten text from IAM dataset

Handwritten text from CVL and CAR dataset
(Real data collected from bank cheques)

Handwritten text from PhPAIS dataset (Real cell-phone
number collected from China post mail images)

Figure 1.5 : Samples of handwritten text. (selected from IAM dataset [86], real bank cheque dataset CVL and CAR [26], and real cell-phone number dataset PhPAIS [128]).

Finally, we would like to briefly introduce the flagship conference in the field of document analysis, *i.e.,* International Conference on Document Analysis and Recognition (ICDAR), where various robust text reading competitions are organised and many benchmark datasets are released. ICDAR is held every two years since 1991. The community provides us an excellent communication platform by organizing competitions, workshops and tutorials. In every ICDAR, hundreds of scientists, researchers and practitioners working on document understanding, historical document analysis, graphic analysis and robust text reading are gathered to present their work, exchange ideas, seek opportunities and discuss the future of related researches. Moreover, the sister conference of ICDAR, *i.e.,* the International conference on Frontiers in Handwriting Recognition (ICFHR), is specially organised for handwriting recognition. Therefore, we can say that ICDAR has contributed a lot to document analysis and has witnessed the development of text detection and recognition.

1.2 Motivation

Last few years have witnessed the explosion of deep learning techniques, especially those developed for computer vision tasks. Meanwhile, solutions to text detection and text recognition have been through a great revolution since 2015, when deep learning was introduced to this area and has since made a great breakthrough.

The detection and recognition pipelines used before 2015 are very complicated and involve sub-tasks like character candidate extraction, non-text filtering, word merging, text segmentation, single character recognition, word rectification, path evaluation, optimal path searching, etc. In these sub-tasks, hand crafted features and carefully designed classifiers play critical roles. According to literature, though these traditional optical character recognition (OCR) techniques have been studied intensively for decades, they can only achieve good performance on scanned and regular texts, even for the commercial OCR software like ABBYY FineReader and Tesseract. As for the complicated scene texts and handwritten texts from real-word applications, their performance is still far from satisfactory.

In contrast, after 2015, deep learning-based models become the dominating solutions to both text detection and text recognition because of their fantastic characteristics presented below.

- Strong representation abilities. Deep networks can extract powerful, adaptive and discriminative features from complex images, and when the training data is adequate, their representation abilities will increase as the network depth and width expanded.
- Flexible structures. Various activation functions (sigmoid, tanh, ReLu, Leaky ReLu, etc.), loss functions (Softmax Loss, Cross Entropy Loss, L1 Loss, L2 Loss, IoU Loss, GAN Loss, etc.), network layers (convolutional layer, deconvolutional layer, pooling layer, upsampling layer, normalization layer, fully connected layer, etc.), network modules (SPP, ASPP, Inception, RoIAlign, LSTM, etc.), network backbones (VGG, ResNet, Xception, GAN, GCN, etc.) and optimizers (Adam, SGD, Adadelta, Amsgrad, etc.) can be flexibly combined according to customers' requirements, and the way to combine them can even be automatically searched via Neural Architecture Searching (NAS) techniques. Moreover, thanks to this flexibility, network sharing and knowledge transfer are allowed among different tasks so that problems like data scarcity and slow convergence can be addressed. For example, backbones pre-trained on the ImageNet dataset are often employed by models for other computer vision tasks including object detection, semantic segmentation, text detection/recognition and so on.
- Convenient architectures. Deep networks integrate their feature extraction module and task-specific module into one unified and end-to-end trainable framework, and the training procedure is automatically performed without manual adjustment. In addition, when the same network is applied for different applications, we just need to format new data and annotations in a particular way and then feed them into the network.

- Well-developed frameworks and technical documents. Many advanced and highly optimized deep learning frameworks and tools like Caffe, Torch, Keras, Theano, Tensorflow, MXNET, etc., and their technical documents are available online. Especially, the most widely used Tensorflow has been updated from v0.12 (static graph version) to v.20 (dynamic graph version) within about five years, and thus is highly complimented by researchers from both industry and academia.

Compared with traditional text detectors and recognisers, the deep learning-based ones are with more compact structures, more robust features, more discriminative classifiers, less pre-/post-processing operations and most importantly, better performances. Therefore, in recent five years, almost all of the state-of-the-art text detectors and recognisers are based on deep learning. In this thesis, given the above observations and advantages of deep networks, we conduct research on text detection and text recognition based on deep learning techniques.

1.3 Related Works

In this section, we firstly give a brief description to convolutional neural network (CNN) and Long Short-term Memory (LSTM), which are the two most basic deep learning techniques for text detection and text recognition. Then, literature review regarding text detection and text recognition are followed. Given the popularity of traditional detectors and recognisers before 2015, and the effectiveness and superiority of the most recent deep learning-based ones, in this part, we will present more details to the latter and less details to the former.

1.3.1 Convolutional Neural Network and Long Short-term Memory for Text Detection and Recognition

Fig. 1.6 shows a typical CNN structure named LeNet5 [56], which is composed of two convolutional layers, two pooling layers, two fully connected layers and one softmax output layer. Here, the softmax layer is designed as a task-specific module, and the other layers are utilized for feature extraction. Especially, the

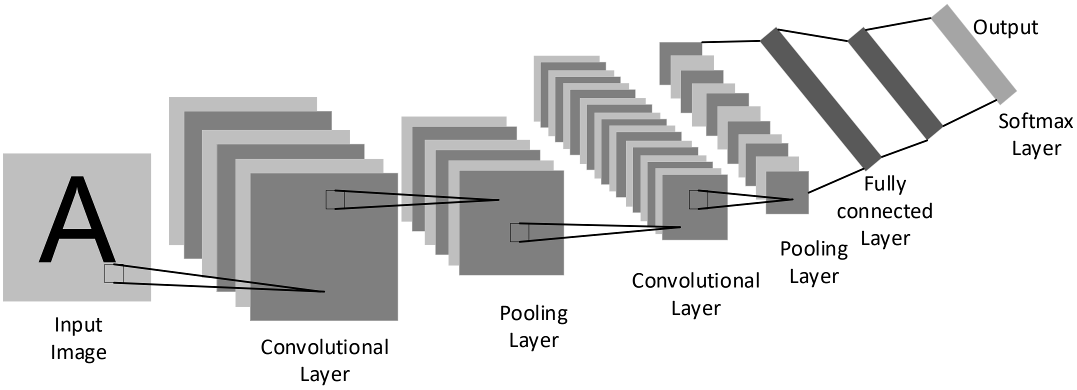


Figure 1.6 : Structure of LeNet5 [56]).

convolutional layers imitate human receptive field mechanism and share kernels among individual locations. Therefore, compared with the fully connected layers, they introduce fewer parameters and are capable of extracting features from inputs with arbitrary sizes.

Given input X of the l^{th} convolutional layer or fully connected layer, *i.e.,* output of the $(l-1)^{th}$ layer, we can calculate output O of the l^{th} layer with the function shown in Eq. 1.1, where W, b and f denote the randomly initialized kernels, bias and activation function, respectively. Note that, in the convolutional layers, $*$ means the convolution operation, while in the fully connected layers, it represents the vector product operation. Pooling layers of CNNs introduce no parameters and they are usually employed to increase networks' invariance towards translation, rotation and scale, and meanwhile, reduce the number of dimensions so that the computation burden can be alleviated. In addition, Batch Normalization (BN) layer proposed in [47] is also critical to CNNs as it can effectively reduce the internal covariate shift of input features. In the existing CNN models, BN layers are often embedded before the convolutional layer and the fully connected layer, and related calculation can be formulated in Eq. 1.2, where x_i and y_i are the input feature maps and output feature maps of the BN layer, and μ and δ^2 can be regarded as the mean and variance of the m samples in current mini-batch, respectively.

$$O = f(X * W + b). \tag{1.1}$$

$$\begin{aligned}
&\mu = \frac{1}{m}\sum_{i=1}^{m} x_i,\\
&\delta^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu)^2,\\
&\widehat{x}_i = \frac{x_i - \mu}{\sqrt{\delta^2 + \epsilon}},\\
&y_i = \gamma \widehat{x}_i + \beta.
\end{aligned} \tag{1.2}$$

In recent years, a large number of variants of the above mentioned layers have been proposed in CNNs, such as atrous convolutional layer, depth separable convolutional layer, RoIAlign layer, etc. By combining these layers, new CNN structures have been developed and achieved promising performances in various computer vision tasks, such as IncepText [141] designed for text detection, as presented in Fig. 1.7.

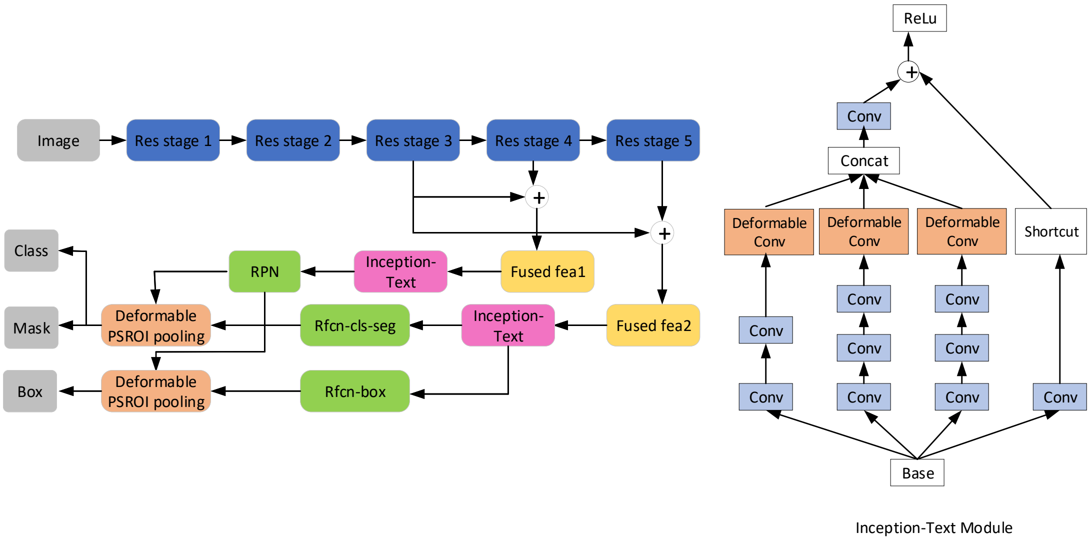
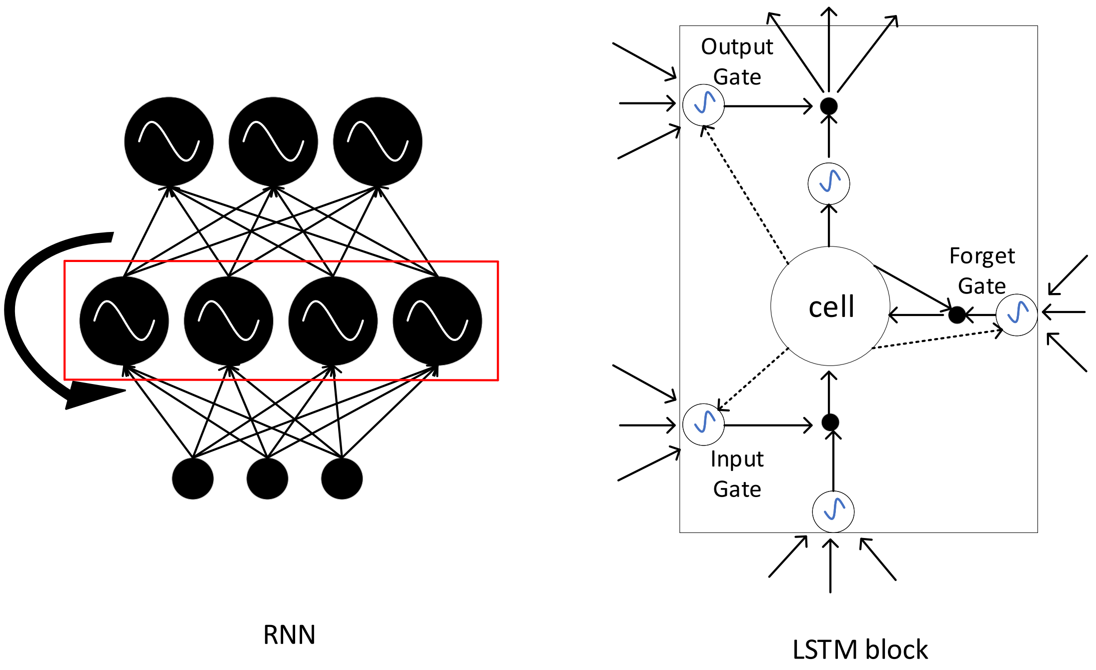


Figure 1.7 : Structure of IncepText [141]).

Recurrent Neural Network (RNN) take activations from last time step as inputs of current time step, as shown in Fig. 1.8, so it can bridge long time delays of inputs and takes historical information into consideration when performing predictions. However, as pointed out in [2], RNN suffers from the long time dependency problem, *i.e.,* it forgets information from the long past. To tackle this problem, LSTM introduces three gates on the base of RNN, *i.e.,* input gate, forget gate and output gate, to store and access information over long time periods. Flow diagram of LSTM is presented in Fig. 1.8, and involved calculations are formulated in Eq. 1.3, where ι, ϕ and ω represent input gate, forget gate and output, respectively, and b_ι^t, b_ϕ^t and b_ω^t denote outputs of corresponding gates. Here, s_c^t and b_c^t are the cell state and output of current time step t.



RNN

LSTM block

Figure 1.8 : Structure of RNN and LSTM [2]).

Inspired by speech recognition and machine translation, text recognition has been widely treated as a sequence-to-sequence problem, where LSTM plays a critical role. However, RNN and LSTM are originally designed for processing stream signals, *e.g.,* audio and sentences, so they take 1-D feature vectors as inputs, as illustrated in Eq. 1.3. To adapt LSTM to text recognition, a straightforward way is to map 2-D feature maps of text images into 1-D space via the flattening or pooling operations. Unfortunately, this will result in the neglect of the valuable spatial and structural information of 2-D text images. To tackle this problem, bi-direction LSTM (bi-LSTM) and multi-dimensional LSTM (MDLSTM) are further proposed in literature to extract information from two directions or four directions with a stack of LSTMs. This strategy works to some extent, but it also introduces extra computations, which linearly increase with the number of LSTM layers. Therefore, as a trade-off between accuracy and efficiency, current text recognisers prefer bi-LSTM over other LSTM variants.

$$
\begin{aligned}
&a_{\iota}^{t} = \textstyle\sum_{i=1}^{I} w_{i\iota} x_{i}^{t} + \sum_{h=1}^{H} w_{h\iota} b_{h}^{t-1} + \sum_{c=1}^{C} w_{c\iota} s_{c}^{t-1},\\
&b_{\iota}^{t} = f(a_{\iota}^{t}),\\
&a_{\phi}^{t} = \textstyle\sum_{i=1}^{I} w_{i\phi} x_{i}^{t} + \sum_{h=1}^{H} w_{h\phi} b_{h}^{t-1} + \sum_{c=1}^{C} w_{c\phi} s_{c}^{t-1},\\
&b_{\phi}^{t} = f(a_{\phi}^{t}),\\
&a_{c}^{t} = \textstyle\sum_{i=1}^{I} w_{ic} x_{i}^{t} + \sum_{h=1}^{H} w_{hc} b_{h}^{t-1},\\
&s_{c}^{t} = b_{\phi}^{t} s_{c}^{t-1} + b_{\iota}^{t} g(a_{c}^{t}),\\
&a_{\omega}^{t} = \textstyle\sum_{i=1}^{I} w_{i\omega} x_{i}^{t} + \sum_{h=1}^{H} w_{h\omega} b_{h}^{t-1} + \sum_{c=1}^{C} w_{c\omega} s_{c}^{t},\\
&b_{\omega}^{t} = f(a_{\omega}^{t}),\\
&b_{c}^{t} = b_{\omega}^{t} h(s_{c}^{t}).
\end{aligned}
\tag{1.3}
$$

1.3.2 Text Detection

In this section, we review literature for text detection. Text detectors applying traditional techniques are very popular before 2015, but their performance are far behind that of the CNN-based ones. Therefore, to follow the new trend, in this part, our focus is put on the CNN-based text detectors.

1.3.2.1 Region-based and CC-based Detectors

Traditional text detectors can be split into two groups: region-based ones and CC-based ones. Region-based detectors utilize sliding windows to localize regions with high confidence to be texts, and the distinct texture of texts, *e.g.,* Wavelet coefficients, Discrete Cosine Transform, etc., make it possible to differentiate text regions from non-text ones. According to literature, region-based detectors are robust to noise, but, to adapt to different text sizes, they often leverage multi-scale operators, resulting in time-consuming systems. In contrast, CC-based detectors assume that pixels belonging to the same CC have similar local properties such as grey level, intensity, stroke width, etc., and characters can be extracted from images by generating CCs. The number of CCs is much less than that of regions, so CC-based detectors are much more efficient than region-based detectors. However, it is a tough task to find a robust CC generator that can deal with blur, skew, low resolution, uneven illumination, etc.

Region-based detectors. Gao et al. [29] obtained numerous regions by sliding pre-defined windows along input images, and extracted 104 features from each of them, including HOG, LBP, number of extended edges, etc. Subsequently, these features were fed into a cascade Real Adaboost classifier to distinguish text regions from non-text ones. Learners of the cascade classifier were re-weighted according to their abilities, and afterwards, candidate regions were re-scored by the re-weighted learners before proceeding to the followed post-processing procedure. Wang et al. [131] employed a 32-by-32 sliding window to create regions, which were then fed into a two-class CNN classifier. The first layer of their CNN was trained with a k-means related unsupervised algorithm, and parameters in this layer were fixed when the whole network was optimized with L_2-SVM classification error. To ease the computation burden, Neumann et al. [92] only took regions containing strokes into consideration, so they conducted stroke detection before any other operations like patch classification, word formulation, etc.

CC-based detectors. The key to the success of CC-based text detectors is finding a robust and effective CC generator. Currently, there are two most widely used CC generators in literature, *i.e.,* Stroke Width Transform (SWT) proposed by Epshtein et al. [27] and MSER proposed by Matas et al. [87]. SWT creates CCs according to the consistency of stroke width, while MSER takes advantage of pixels' intensity similarity. Specifically, SWT calculates stroke widths for individual pixels by measuring the distances of opposite point pairs from edges detected by the well-known Canny edge detector [50], while MSER can be understood from the view of image binarization. If a threshold is continuously changed from 0 to 255, a large amount of components, so-called extremal regions, can be obtained from the binarization results of input images. MSER enumerates these extremal regions and selects an affinely invariant subset from them. According to literature, among all of the traditional text detectors, ones based on SWT and MSER achieve state-of-the-arts and outperform others significantly.

Yao et al. [143] employed SWT as CC generator, and then classified CCs with a two-stage coarse-to-fine classification framework. Bai et al. [4] also made use of characteristics of strokes to produce CCs, but different from SWT, they calculated text confidence maps according to the density of pairwise edges and the consistency of stroke width. Yin et al. [148], the winner of ICDAR 2013 Robust reading competition [53], experimentally proved that MSER was more effective than SWT. In their work, MSERs were detected and pruned firstly to generate character candidates. The prune algorithm consisted of two sub-steps, *i.e.,* linear reduction and tree accumulation. Then, a single-link clustering procedure was performed to construct these candidates into groups that corresponded to words or text lines. In this stage, distance metric learning and features like interval, width and height differences as well as colour differences were adopted. Afterwards, a text classifier was employed to eliminate false positives. Finally, this work was extended to handle multi-oriented text detection via a forward-backward algorithm. Neumann et al. [94] also held the view that Extremal Regions (ERs) were more robust to colour, blur, uneven illumination, etc., than other CC generator, so they proposed a real-time scene text detector on the base of ERs. However, the way they selected character candidates from ERs was different from that of MSER. In particular, they calculated a group of incrementally computed descriptors including area, perimeter, Euler number, etc., for individual ERs, and fed these descriptors into a Real AdaBoost classifier to obtain text probabilities. Only ERs with locally maximal probabilities were kept and processed by subsequent steps. Afterwards, SVM classifier together with another group of computationally expensive features, such as hole area ratio, convex hull ratio, etc., were utilized to perform further ER selection. Chen et al. [10] took MSERs with enhanced edges as their character candidates. Shi et al. [110] built a graph model on the base of MSERs, and employed the max-flow/min-cut algorithm to optimize it. Neumann et al. [93] designed a variant of MSER, *i.e.,* MSER++, for scene text detection, and proposed an efficient pruning algorithm for the exhaustive search so that the system could work in real-time. Yin et al. [149] utilized geometry-based grouping and AdaBoost classifier to extract characters from MSERs.

In summary, the pipelines of region-based detectors and CC-based detectors are very complicated and researchers working on this area have concentrated on designing effective and efficient hand-crafted features, single character classifiers and pruning strategies. We suggest readers refer to [145] and [159] for more comprehensive surveys on detectors applying traditional techniques.

1.3.2.2 CNN-based Detectors

As clarified in [81], text detection and text recognition have entered a deep learning era. CNN-based detectors and recognisers integrate feature extraction and task-specific module into one end-to-end trainable framework, so the reading systems are becoming more and more compact. Moreover, compared with traditional detectors, CNN-based ones only require some simple data augmentation operations like resizing, cropping, flipping, etc., in the pre-processing stage to boost detection performances, and most of the CNN-based detectors only need the Non-maximum Suppression (NMS) algorithm in the post-processing stage to remove redundant bounding boxes.

According to literature, currently, there are two mainstream branches for CNN-based detectors. Models in the first branch cast text detection to the semantic segmentation problem and perform pixel-level predictions. The prediction results of individual pixels may refer to text confidences, distances to edges of corresponding bounding boxes, links to adjacent pixels, angles of segments or other pre-defined concepts. In contrast, models from the second branch treat texts as a kind of particular objects, and utilize general object detection pipeline to handle this task. In particular, they firstly generate a large number of text proposals, and then perform classification and regression to refine them and, meanwhile, filter out those without texts. Note that, when traditional techniques are used, text detection, semantic segmentation and object detection are studied separately because traditional models rely heavily on task-oriented features and frameworks, while after the introduction of deep learning, connections among these tasks become stronger and ideas as well as network structures are often shared by them since deep networks automatically learn adaptive features under the guidance of the task-specific modules.

Text Detectors Motivated by Semantic Segmentation. The popularity of deep learning in semantic segmentation starts from the propose of Fully Convolutional Network (FCN) [80], which is the first network to perform pixel-level predictions. Thereafter, a series of FCN variants are reported and achieve state-of-the-arts in literature. Inspired by these advanced semantic segmentation models, FCN-derived text detectors are also widely explored since 2015. For example, Yao et al. [144] designed a single FCN to predict maps corresponding to text regions, characters and linking directions of characters. Since character-level annotations were required in this mode, word partition had to be done on images from datasets that only provided word-level ground truth. In the inference stage, a formulation step was designed to obtain final rotated bounding boxes. Zhang et al. [153] also made use of salience maps generated by FCN, but their text lines were produced by combining salience maps and MSERs. Wu et al. [136] proposed border learning in this work, where pixels were classified into text, non-text and borders of text areas at the output layer of FCN.

Recently, a more simplified pipeline was explored by many state-of-the-art approaches. This pipeline predicted offsets of bounding boxes in a more straightforward way. For instance, EAST proposed by Zhou et al. [156] employed PVANet [55] to perform pixel-level prediction and produce score and geometry maps. The score map indicated how likely current pixel belonged to text, and the geometry maps (four maps for rotated boxes or eight maps for quadrangles) were used to restore bounding boxes for individual pixels. Finally, redundant bounding boxes were removed by NMS. He et al. [44] named the offset regression from a given point as direct regression, and analysed the drawback of indirect regression, which was widely used in Faster-RCNN [98] and Single Shot Multibox Detector (SSD) [73]. Afterwards, they designed a simple and effective detector for multi-oriented text detection and proposed a new NMS, *i.e.,* recalled NMS, to remove redundant bounding boxes. PixleLink proposed in [24] took VGG as its backbone, and predicted score maps and links for individual pixels at the header layer. Given a pixel, its links were presented by eight maps with respect to connections to its eight adjacent neighbours. In [123], Wang et al. proposed Instance Transformation Network (ITN) to build a translational variant scene text detector. In this work, an in-network transformation embedding module was set to learn a geometry-aware representation and an affine matrix, which were then employed by a multi-task module. The multi-task module performed classification, coordinate regression and transformation regression, so compared with other models, additional six parameters needed to be predicted for individual pixels in ITN. Long et al. [82] treated text with arbitrary shape, *i.e.,* curved text, as a sequence of oriented and overlapped disks, which could be represented by radius and orientations. Therefore, they proposed TextSnake, an FCN-based scene text detector that produced score maps for text regions, text centres as well as disks' geometry attributes regarding to radius, $cos\theta$ and $sin\theta$, where θ denoted disks' orientations. To reconstruct text regions from predicted maps, a striding algorithm was designed in their post-processing stage. Xue et al. [139] proposed to detect text borders in the view of semantic segmentation, and designed a bootstrapping algorithm to augment their training data.

Text Detectors Derived from Object detection. Currently, one-stage detectors and two-stage detectors are the two best solutions to general object detection. The former are known for their efficiency, while the latter are popular for their effectiveness. SSD [73] and Faster R-CNN [98] are the most well-known representatives of these two branches, and have played a key role in object detection. SSD produces prediction results directly with one single network, while faster R-CNN combines Region Proposal Network (RPN) and Fast R-CNN [33] to perform classification and regression twice with a two-stage framework.

Scene text detectors derived from object detection could be motivated by either SSD or Faster R-CNN. For example, inspired by SSD, He et al. [43] proposed an attention-equipped single shot text detector, where a hierarchical Inception module and a text attention module were combined to aggregate deep features from multiple scales and suppress background pixels. At the output layer, for each default word box, a softmax classifier and a regression module were employed to predict its text confidence and regression parameters. TextBoxes proposed in [64] was also motivated by SSD. This network took VGG [113] as its backbone, and predicted a 72-D vector for each location at the output layer, *i.e.,* 2-D scores and 4-D offsets for 12 default boxes. To deal with text with arbitrary orientations, Liao et al. [63] further improved TextBoxes by predicting additional offsets for rectangle boxes (4-D) or quadrilateral bounding boxes (8-D) at the text-box layer. Shi et al. [106] held the idea that the existing object detection-based methods might miss some long words or text lines due to the limitation of default anchor setting. Therefore, they proposed to predict bounding boxes for segments, *i.e.,* portions of words, and related links, which were used to indicate whether two segments belonged to the same word or not. Classification and regression were two indispensable modules in most state-of-the-art approaches, and they were usually built on the same deep features. However, between these two modules, only the regression was rotation sensitive. Therefore, in [66], a group of rotation-sensitive features and a group of rotation-invariant features were specially extracted for regression and classification, respectively. By doing so, a more accurate regression could be achieved for long and thin texts. The existing SSD-inspired scene text detectors usually exploited FCN for prediction, but in [100], Rong utilized an LSTM-based decoder to sequentially predict bounding boxes and corresponding confidence scores. Therefore, their model was able to used for both scene text detection and context reasoning text retrieval.

On the other hand, many scene text detectors were inspired by Faster R-CNN, such as R^2CNN proposed by Jiang et al. [49]. In R^2CNN, a batch of axis-aligned text boxes were firstly generated by Region Proposal Network (RPN). Then, a pooling layer with multiple pooling sizes was assembled to convert feature maps into vectors before the prediction layer, which was designed to calculate text/non-text scores, axis-aligned box coordinates and inclined box coordinates for axis-aligned text boxes. Liu et al. [77] proposed a curved text detector under the coarse-to-fine framework. In their model, ResNet [42] was used as the backbone of entire network, and RPN was used to roughly detect text regions with default rectangular anchors. Subsequently, an RNN-based regression module was designed to predict the offsets of curved locating points. RPN produced horizontal proposals, but texts presented in scene images were usually with arbitrary orientations. To improve the robustness of current models, Ma et al. [85] proposed Rotated Region Proposal Network (RRPN) to generate rotated proposals. Moreover, they also proposed rotated RoI (Region of Interest) pooling, skew IoU computation and skew NMS to work with their RRPN. Zhang et al. [152] thought that traditional RPN module only applied 1×1 convolution, which was insufficient for text detection. Therefore, they presented a feature enhancement network to fuse task-specific features, high-level features and low-level features for region proposals. Tian et al. [121] proposed a Connectionist Text Proposal Network (CTPN) for accurate text detection. Different from traditional FCN style RPN, CTPN employed Bi-LSTM and fully connected layer for score, coordinates and offsets prediction. Dai et al. [23] designed a Fused Text Segmentation Network, where two groups of fused feature maps were generated for region proposals and text instance predictions, respectively. Zhu et al. [158] incorporated a vertical proposal mechanism in RPN to avoid proposal classification, and achieved good performance on horizontal text detection.

Mask R-CNN [40] performed pixel-level prediction inside of text proposals obtained by Faster R-CNN. According to literature, scene text detectors based on Mask R-CNN have achieved state-of-the-arts among the existing works. For example, SPCNET proposed in [137] applied MASK-RCNN with Feature Pyramid Network (FPN) to scene text detection. To reduce false positives and improve accuracies of predicted confidence scores, a re-score mechanism was designed in this work. Huang et al. [46] proposed Pyramid Attention Network (PAN) to improve performance of Mask R-CNN. According to their reported results, false positives posed by text-like regions were able to effectively suppressed by PAN. Moreover, Mask R-CNN achieved promising performance in [70] by augmenting training samples via cropping operation, and its performance was able to further improved by replacing traditional masks with newly designed soft masks.

Text Detectors Combining Two Branches. According to literature, object detection-based methods are usually troubled by long texts and texts with arbitrary directions, while semantic segmentation-based ones suffer from relatively complicated post-processing steps. To take advantage of both branches and, meanwhile, avoid their shortcomings, Lyu et al. [84] proposed a network by combining ideas from both branches. In particular, a position sensitive segmentation module and a corner detection module were designed to predict text score maps and corner locations of related bounding boxes, respectively. In this work, NMS was also applied in the post-processing stage to reduce redundant bounding boxes. Mask Textspotter [83] proposed for curved text detection and recognition leveraged an Faster R-CNN branch to detect text areas. Subsequently, a pixel-level word/character segmentation branch was carefully designed for recognition. He et al. [39] divided the detection task into two sub-steps in their work, *i.e.,* using a multi-scale FCN to detect text blocks from input image, and segmenting text lines by predicting text centre lines via instance segmentation. Pixel-Anchor network proposed by Li et al. [61] took ResNet-50 as its backbone, and performed both pixel-level segmentation and anchor-based detection upon feature maps extracted by the ResNet-50. Apart from rotated boxes, the segmentation module also produced some heat maps, which were regarded as attentions and were fed into subsequent anchor module to generated more accurate predictions. The anchor module was modified from SSD, and to adapt to long text regions, it was equipped with some adaptive predictor layers. Yang et al. [141] also employed ResNet-50 as their backbone, but, different from [61], ResNet-50 used in Yang's work was assembled with the effective deformable convolution layer [22] and deformable PSROI pooling layer [22].

1.3.3 Text Recognition

Since scene texts and handwritten texts have their unique characteristics, corresponding traditional recognisers have been developed under different pipelines. In particular, scene text recognisers can be broken into text segmentation, single character recognition and lexicon-based or language model-based word rectification. Especially, text segmentation, *i.e.,* discriminating foreground text pixels from background pixels with binarization or graph models, is the bottleneck of such recognisers [1] because of the complicated backgrounds and unconstrained imaging conditions of scene images. On the other hand, to handle the intractable touching characters, handwritten text recognisers usually over-segment text images into components, *i.e.,* characters or portion of characters, and then combine these components to form a candidate lattice. Subsequently, nodes in the lattice are recognized as single characters and individual paths of the lattice are evaluated by considering multiple information like classification scores, geometric contexts, linguistic contexts, etc. Finally, based on the evaluation results, the sequential outputs are inferred from the optimal path searched from the lattice.

Recently, with the introduction of deep learning techniques, scene/handwritten text recognition is widely treated as a sequence-to-sequence prediction problem, and to avoid the troublesome segmentation prerequisite, segmentation-free framework is studied intensively. Inspired by speech recognition and machine translation, LSTM has been playing a critical role in the most recent recognisers. According to their decoder modules, these recognisers can be grouped into the Connectionist Temporal Classification-based (CTC-based) [35] ones and the attention mechanism-based [3] ones. Note that, for models from both groups, LSTM is mainly used for the following three purposes, *i.e.,* feature encoding (models in both categories), frame-level prediction (CTC-based models) and sequential translation (attention mechanism-based models). Though the performance of the LSTM-based recognisers has surpassed that of traditional recognisers significantly, some researchers [65, 127] claimed that LSTM-based recognisers could only achieve promising performances on horizontal or near horizontal texts because LSTM takes

1-D feature vectors as inputs, and thus could not fully leverage the spatial information of 2-D text images. Therefore, in literature, some LSTM-free recognisers were also explored.

1.3.3.1 Traditional Segmentation-based Recognisers

Recognisers for Scene Text. Phan et al. [96] proposed a character segmentation method based on gradient vector flow. They cast character segmentation as a minimum cost path searching problem. Nomura et al. [95] designed an adaptive morphological model for degraded scene text segmentation. Sheshadri et al. [105] utilized exemplar SVM for scene character recognition. Tian et al. [120] proposed Co-HOG and ConvCo-HOG for the recognition of segmented single characters by utilizing the co-occurrence of orientation pairs of neighbouring pixels. Shi et al. [111] designed a tree-structured model to detect locations of individual characters, and then utilized CRF to infer character classes of individual locations. Considering difficulties of text segmentation, Mishra et al. [89] leveraged sliding windows to roughly segment scene text images into patches, and then employed the bottom-up and top-down cues for subsequent text recognition. Ye et al. [145] proved a comprehensive survey on methods following this pipeline.

Recognisers for Handwritten Text. Wang et al. [130] proposed an effective Chinese HTR system under the over-segmentation framework. Firstly, a CC-based segmentation method [68] was used to segment input text images into primitive segments, which were then formed into a candidate lattice. Afterwords, character classification scores, geometric contexts and linguistic contexts were combined from the Bayesian decision perspective to calculate the overall score of each potential path. Finally, a refined beam search strategy was designed to infer the optimal paths and produce final recognition results. Wang et al. [122] designed a real-time Chinese HTR system by performing dynamic text line segmentation and character over-segmentation. In this work, linguistic contexts and geometric contexts were recomputed when a new character candidate was generated. Lee et al. [58] presented a binary segmentation algorithm for cursive English handwriting recognition. In this work, word images were iteratively cut into sub-images until pre-defined conditions were satisfied. Zhou et al. [154] built a semi-Markov Conditional Random Fields (semi-CRFs) upon the candidate lattice for Chinese/Japanese HTR, and designed a negative log-likelihood loss to optimize related parameters. As claimed in [122], language model with higher order than bi-gram was hard to integrate into CRF, but in [154], a trigram language model was successfully applied. To ease the computation burden introduced by language model, a forward-backward pruning algorithm was further proposed in this work. On the basis of [154], Zhou et al. [155] proposed an alternative minimum-risk training strategy for optimizing their semi-CRF. From the comparison with other three non-uniform cost functions, effectiveness of their proposed training strategy was proved. Wang et al. [129] evaluated performances of various language models under the over-segmentation framework, including character-level n-gram models, word-level n-gram models and hybrid n-gram models. As mentioned before, n-gram statistic language models played important roles in HTR, but because of the problem of data sparseness, high order language models were hard to use. To address this issue, Wu et al. [135] proposed a Feedforward Neural Network Language Model (FNNLM) to project history characters into a continuous space and produce sequence probabilities for input images. Keysers et al. [54] gave an brief introduction to Google's fast and high-accuracy HTR system, by which 22 scripts and 97 languages were supported. Thanks to the speed accelerating techniques used in this work, such as segment pruning, lattice edge pruning, edge factor pruning, beam search pruning, etc., the proposed system was able to run in mobile devices or cloud. Specifically, 5 percent of writing areas were re-sampled in the pre-processing stage to generate inputs for the following steps. Then, a slope correction step was performed to handle skew text lines. In the character classification module, a heuristic classifier and an NN classifier were utilized to obtain a set of high recall character hypotheses, which were then formed into a segmentation lattice according to their time order and spatial order. Afterwards, to decode the lattice, a 9-gram character language model pruned by a stupid-backoff [9] entropy was built upon a large corpus and a word-based probabilistic finite automaton was employed carefully.

1.3.3.2 Segmentation-free Recognisers Based on CNN and LSTM

CTC-based Recognisers. CTC was proposed by Graves et al. [35]. Given the frame-level prediction $Y = \{y_1, y_2, ..., y_n\}$ of input images, CTC calculates the probability $P(L|Y)$ of a label sequence L via a forward-backward algorithm with considering all of the possible paths. Here, the frame-level prediction is usually generated by RNN or its variant LSTM. In Graves's work [35, 36, 79], CTC was proved to be effective and efficient in speech recognition and text recognition. From their experimental comparison, recognisers applying CTC achieved superior performance over traditional HMM-based ones. CRNN proposed by Shi et al. [107] was the first to successfully integrate CNN, LSTM and CTC into one unified network in the field of scene text recognition. It freed recognisers from character-level annotations by casting the recognition problem to a sequence-to-sequence problem. Before that, CNN was only used for single character classification in DCNN models and LSTM took raw images or hand-crafted features like HOG as inputs [2, 37]. In CRNN, convolutional layers were employed to extract deep features from input images, the followed bi-LSTM layers were responsible for encoding frame-level features and producing frame-level predictions, and CTC was embedded at the end of the network to conduct sequential transcription. START-NET proposed in [75] also followed the CNN-LSTM-CTC architecture, but different from RCNN, it utilized the well-known residue convolutional blocks for feature extraction. In addition, to deal with images with distortion, Spatial Transformer Network (STN) was assembled at the beginning of START-NET to rectify distorted texts into ones with more canonical appearances. As mentioned before, scene texts were usually surrounded by complicated backgrounds, which posed huge challenges to the recognition task. To help models focus on foreground areas, Gao et al. [32] proposed an attention-boosted dense chain network. They also claimed that RNN produced prediction results frame-by-frame, and this was not suitable for parallel computation. Therefore, in [31], the RNN module was replaced by a convolutional module. According to their comparison results, CNN-based sequence modelling seemed more efficient, but the RNN-based one was able to achieve better accuracy. Yin et al. [147] also kept the view that RNN/LSTM-based models were harder to be trained because of the gradient vanishing problem, and CNN-based frame-level prediction was a better choice. Therefore, before CTC, a sling window layer was designed in their recogniser to extract deep features and perform prediction.

Messina et al. [88] took advantages of three Multi-Dimensional Long-Short Term Memory (MDLSTM) layers when extracting features from input raw images. Different from bi-LSTM used in [36, 79], MDLSTM took features from four directions into consideration so that more accurate predictions could be produced. Moreover, to generate frame-level predictions required by CTC, some fully connected layers and a softmax classifier were assembled between MDLSTM and CTC in this work. Xie et al. [138] proposed a Fully Convolutional Recurrent Network (FCRN) for online Chinese HTR. Bi-LSTM and CTC were also employed in this work for frame-level prediction and sequential transcription, just as [36, 79] did. But instead of still images, FCRN took pen-tip trajectory as inputs. To convert variable-length trajectory into signature feature maps while preserving the analytic and geometric characteristics of online paths, a path-signature layer and a fully convolutional network were designed before bi-LSTM. FCRN was then extended to a more powerful version, *i.e.,* MC-FCRN, by learning multi-spatial-context information via FCN and designing an implicit language model. Sun et al. [117] utilized a deep RNN to learn an mapping function that was able to project input trajectories to strings.

Attention Mechanism-based Recognisers. The idea of attention mechanism is borrowed from speech recognition [20] and machine translation [3], and it is usually combined with Gated Recurrent Unit (GRU) [19] or LSTM to map input deep features to expected sequential outputs. An attention mechanism-based sequential transcription module is auto-regressive and does not require frame-level predictions. Therefore, compared with CTC-based recognition, attention mechanism-based ones are more popular in recent works and achieve better recognition performance. For example, Lee et al. [57] employed attention-equipped RNN for sequential transcription in their R^2AM, where a recursive recurrent network was designed for more effective and efficient feature extraction. The same as START-

NET [75], RARE proposed in [108] also employed STN to rectify skewed text images, but instead of CTC, they combined attention mechanism and GRU to directly infer sequential outputs from input feature maps. As clarified in [6], irregularly shaped art texts presented frequently in our daily life, especially perspective texts and curved texts, which had posed huge challenges to scene text recognition. To tackle this problem, Cheng et al. [16] combined the attention mechanism with a specially designed CNN in their recogniser, *i.e.,* AON, where a sibling CNN branch and a shared CNN branch were designed to extract character placement features and features from both horizontal and vertical directions, respectively. Afterwords, these features were combined and filtered with a filter gate before being fed into the subsequent attention module. Recogniser proposed by Yang et al. [142] was aimed to handle distorted and curved texts. It consisted of three components, *i.e.,* a basic feature extraction network, a pixel-wise character mask prediction network and an attention-equipped RNN sequential transcription network. The predicted character masks were expected to suppress noise and help to learn text specific patterns by capturing contextual information. In addition, an attention alignment loss was designed in this work to help find proper attention areas at the early training stage because network's parameters were randomly initialized and thus the attention-based model might be ineffective at the beginning. Since most existing datasets did not provide ground truth for attentions, a novel attention construction algorithm was presented in this work. Char-Net [74] was short for Character-Aware Neural Network, where a Hierarchical Attention Mechanism (HAM) was designed to detect and rectify individual characters. HAM had two layers, *i.e.,* a RoIWarp layer used to recurrently extract character regions and a character-level attention layer used to remove distortion and further encode character regions. ASTER [109] was an enhanced version of RARE [108]. It also consisted of a STN-based rectification network and an attention mechanism-based standard recogniser. ASTER achieved a significant improvement over RARE regarding rectification performance because a more effective Thin-Plate-Spline (TPS)-based STN was adopted to handle various distortions in this work. Besides, ASTER utilized non-linear activation in the localization network to keep the model from gradient vanishing and speed up the training convergence, and it also took advantage of dependencies from both directions by replacing attention-GRU with attention-LSTM.

Sueiras et al. [116] employed a similar recogniser as ASTER [109] to read handwritten texts from IAM (English) and RIMES (French) datasets. In their model, input word images were firstly cut into patches and converted into feature vectors by a LeNet-5 network. Then, an LSTM encoder and an attention-LSTM decoder were applied to further encode features and perform sequential transcription. Bluche et al. [8] designed a handwritten paragraph recogniser, where multiple standard MDLSTM layers were leveraged to extract feature maps from raw images, and an attention-equipped MDLSTM layer was set before LSTM and the softmax predictor.

Though attention mechanism has been widely adopted in the existing state-of-the-art recognisers, it has some inherent shortcomings. For example, Bai et al. [5] pointed out that exiting attention-based recognisers failed to align ground truth strings with attention’ s probability outputs, and this confused and misled the training process of related networks. To tackle this problem, they proposed Edit Probability (EP), which took possible occurrences of missing characters and superfluous characters into consideration when estimating the probability of generating a string from network’ s outputs. In particular, given an input image I and network's parameters Θ, EP calculated probability $EP(T|I;\Theta)$ for string T by summing probabilities of all potential edit paths. Cheng et al. [15] proposed the ‘attention drift’ problem of current attention mechanism, *i.e.,* misalignment of predicted characters and right feature areas. To address this issue, a focusing network was assembled in their FAN to assist the attention network. Recently, Li et al. [59] claimed that traditional attention mechanism was not able to produce accurate attention predictions, thus the recognition performance on irregular text images was largely compromised. To ease this problem, they designed a 2-D attention module, where one LSTM was used to produce holistic features by encoding feature maps column-by-column, and another LSTM was employed as usual to general sequential outputs.

Recognisers from Other Perspectives. Considering drawbacks of CTC-based and attention mechanism-based recognisers, such as slow convergence during training procedure, neglect of spatial information, etc., researchers have also explored other solutions to scene text recognition from other perspectives. For example, SqueezedText proposed by Liu et al. [78] utilized an efficient binary convolutional encoder-decoder convolution network (B-CEDNet) to generate silence maps with respect to individual character classes. Then, a Bi-RNN back-end was followed to correct detection errors and conduct classification. LSTM connected input features in a fully connected way, so 2-D feature maps have to be flattened or pooled into 1-D space before proceeding to LSTM. This would result in neglect of spatial information of scene images. To tackle this problem, Liao et al. [65] proposed to recognize scene text from 2-D perspective. Concretely, they exploited an encoder-decoder FCN, which was equipped with some deformable convolutional layers, to map input images to character confidence maps, and then employed a word formation module to infer sequential outputs from these maps. Masks of character centres were generated at the encoder stage of FCN and, and during decoder stage, these masks were combined with feature maps to suppress irrelevant background pixels and highlight foreground pixels.

Su et al. [115] proposed an HMM-based handwritten text recognition system in their work, where an embedded Baum-Welch algorithm was used to train their HMM by combining with observation sequences obtained from input images. Choudhury et al. [21] designed a set of sinusoidal parameter-based features in their HTR model. These features were sent to a GMM-HMM classifier for inference, and in the prediction stage, likelihood of each lexicon word was calculated by a Viterbi algorithm. Though HMM achieved promising performance in sequence prediction, Schenk et al. [104] pointed out that training of HMM was discriminative. Therefore, to tackle this problem, they proposed two NN-HMM hybrid recognisers. Particularly, an NN classifier was employed in the first recogniser to generate probability outputs, which were then fed into a subsequent HMM module for sequential transcription. In contrast, the second recogniser extracted deep features with NN, and exploited a Principal Components Analysis (PCA) to reduce dimensions of extracted features. Afterwards, the reduced features were sent to a GMM-HMM classifier for inference. Wang et al. [132] also designed a hybrid NN-HMM HTR system under the Bayesian framework. In their work, language model was combined with GMM-HMM, DNN-HMM and DCNN-HMM via a Weighted Finite-State Transducer (WFST)-based decoder to produce sequential outputs. Wang et al. [128] proposed a complete CNN model for HTR, and achieved promising performance on realistic digital string images, which were obtained from real-world bank cheques. Images used to train their network were with arbitrary-sizes, even for those from same batch. To resolve conflicts between arbitrary image sizes and fully connected layers, SPP was adopted and modified in this work to convert feature maps to fixed-length vectors.

1.4 Evaluation Metrics

There are some popular evaluation metrics in the field of text detection and text recognition. To better understand works presented in this thesis, we will give a brief introduction to these widely used metrics below.

1.4.1 Text Detection

Recall (R), Precision (P), F-measure (F) and Average Precision (AP) are the four most well-known scene text detection metrics, as formulated in Eq. 1.4, where N_{corr_det}, N_{det} and N_{gt} denote the number of correctly detected regions, total number of detected regions and total number of ground truth regions, respectively. A region M is regarded as correctly detected when a target region $\widehat{M}$ satisfying $IoU(M, \widehat{M}) > Thred$ can be found from the ground truth set. Here, $Thred$ is a pre-defined threshold. The calculation methods of R, P and F are proposed by Wolf et al. [134] for ICDAR scene text reading competitions held before 2015. According to [134], an evaluation tool built upon these three metrics, *i.e.,* DetEval (`https://perso.liris.cnrs.fr/christian.wolf/software/deteval/index.html`), is made publicly available by the ICDAR community to benefit researchers working on this area. Note that submissions to ICDAR competitions are usually ranked according to the F-measure.

$$\begin{aligned}
R &= \frac{N_{corr\_det}}{N_{gt}},\\
P &= \frac{N_{corr\_det}}{N_{det}},\\
F &= 2\frac{R \cdot P}{R + P},\\
AP &= \frac{1}{11}\sum_{r \in 0,0.1,\ldots,1} Pinterp(r),\\
Pinterp(r) &= \max_{\widetilde{r}:\widetilde{r} \geq r} p(\widetilde{r}).
\end{aligned} \tag{1.4}$$

From 2015 onwards, following the standard practice in general object detection, AP proposed in [28] is used as the basic metric to rank submissions. As defined in Eq. 1.4, if we equally divide the recall value from 0 to 1 into 11 levels, AP equals to the mean precision value of all recall levels. Note that $p(\widetilde{r})$ represents the measured precision at recall $\widetilde{r}$. Therefore, AP can be regarded as a summary of the precision/recall curve.

1.4.2 Text Recognition

For text recognition, mean edit distance (soft metric) and word-level precision (hard metric) are the two most popular metrics in literature. Especially, edit distance with equally weighted operations (insertion, deletion and replacement) and case-insensitive word accuracy are used as ranking metrics in ICDAR 2015 incidental scene text recognition competition and ICDAR 2017 COCO-text recognition competition, respectively. Mean edit distance (MED) and word-level precision (WR) can be formulated as Eq. 1.5, where N_{corr_reg} and N_{str} are the number of correctly recognized strings and the total number of strings to be predicted, and ED is the edit distance between i^{th} ground truth string s^i_{gt} and corresponding predicted string s_{pr}^i.

$$
\begin{aligned}
WR &= \frac{N_{corr\_reg}}{N_{str}}, \\
MED &= \frac{\sum_{i=1}^{N_{str}} ED(s_{gt}^i, s_{pr}^i)}{N_{str}}.
\end{aligned}
\tag{1.5}
$$

1.5 Contributions and Thesis Organisation

In this thesis, we research text detection and recognition based on CNN and LSTM. Our main contributions are presented below.

- We propose a multi-ASPP assembled DeepText network to detect multi-oriented texts from the perspective of image segmentation. The existing detectors usually utilize VGG and ResNet as their backbones, but some other powerful network structures like Xception proposed in DeepLab V3+ [14] have not been explored. Therefore, in this work, we take DeepLab V3+ as our base model and perform pixel-level predictions regarding text scores and geometry information of related bounding boxes. In addition, to improve the recall rate of small text areas, we propose to insert multiple Atrous Spatial Pyramid Pooling (ASPP) layers to the network after feature maps with different resolutions. Moreover, we also employ multiple auxiliary Intersection-over-Union (IoU) losses and auxiliary connections to assist the network training and enhance the discrimination ability of lower encoding layers.

- We propose GMask R-CNN, *i.e.,* Mask R-CNN [40] with global text context, for multi-lingual multi-oriented text detection by following the practice in general object detection. In particular, we conduct score prediction and geometry regression on a large number of anchors, *i.e.,* text proposals, generated by RPN. As we all know, RoIAlign used for extracting fixed-length feature vectors from arbitrary-size proposals is the grantee of efficiency when handling thousands of potential text areas. However, as pointed out in [137], this strategy suffers from false positives and inaccurate segmentation results caused by the lack of context information clues. To tackle this problem, in this work, we design a global mask module between FPN and RPN to perform text predictions with considered global text context, and enhance feature maps with the predicted results before forwarding them to the subsequent steps.

- We propose ReELFA, *i.e.,* Recogniser with Encoded Location and Focused Attention, for scene text recognition by analysing drawbacks of the widely used LSTM and attention mechanism. Scene text recognition has recently been widely treated as a sequence-to-sequence prediction problem, where traditional fully-connected-LSTM (FC-LSTM) and attention mechanism play critical roles. However, FC-LSTM takes 1-D feature vectors as inputs, resulting in severe damage of the valuable spatial and structural information of 2-D text images, and the attention mechanism suffers from the 'attention drift' problem, *i.e.,* networks fail to align attentions on proper feature areas. To ease these problems, in ReELFA, we design an encoded location module to indicate spatial relationships of pixels and a focused attention module to help align attentions on proper feature areas.
- We cast scene text recognition to a spatio-temporal prediction problem and propose FACLSTM, *i.e.,* ConvLSTM with Focused Attention, to address this issue. In FACLSTM, we retain the spatial information of text images by utilizing convolution LSTM (ConvLSTM), where all of the input-to-state and state-to-state transitions are conducted on 2-D feature maps. Furthermore, to take advantage of the attention mechanism, we propose a strategy to harmoniously incorporate it into ConvLSTM via the convolutional operations.
- We propose a sequence labelling convolutional network named CFASPP and apply it to handwritten text recognition. Touching characters are always the bottleneck of handwritten text recognition. To handle such characters, the existing recognisers developed under the over-segmentation framework and segmentation-free framework often conduct patch-level predictions by combining over-segmented components and applying LSTM to sliding windows.

However, patch-level predictions suffer from limited network views, and thus cannot achieve precise predictions. On the other hand, Convolutional Neural Networks (CNNs) with fully connected layers have shown great potentials in various computer vision tasks, but they are not suitable for arbitrary-length string recognition because both the inputs and outputs of the fully connected layers are required to have fixed sizes. In this work, we design a flexible Spatial Pyramid Pooling (FSPP) mediate layer to convert arbitrary-size feature maps to fixed-length feature vectors so that CNN without LSTM can directly transcribe strings from the whole text images. Moreover, by combining with our newly designed connection method, the proposed FSPP layer is able to extract more adaptive features for text images according to their aspect ratios.

This thesis is organised as follows. Firstly, we present our two text detectors, *i.e.,* DeepText and GMask R-CNN, in Chapter 2 and Chapter 3, respectively. Then, details of our scene text recognisers, *i.e.,* ReELFA and FACLSTM, are described in Chapter 4 and Chapter 5, respectively. Afterwards, Chapter 6 shows our sequence labelling network named CFAPP and how we apply it to handwritten text recognition. Finally, a brief summary of this thesis and our recommendation for future works are drawn in chapter 7.

Chapter 2

Multi-ASPP Assembled DeepLab for Multi-oriented Text Detection

Text detection is the first step of a scene text reading system, and its performance has a great impact on the subsequent recognition steps. Therefore, in this chapter, we investigate how to detect texts from scene images. As mentioned previously, in the deep learning era, the issue of text detection can be addressed from the perspective of image segmentation by performing pixel-level predictions, and detectors following this pipeline are more robust to long texts. Given this, in this work, we design our network named DeepText [126] based on DeepLab V3+, which is one of the most well-known image segmentation models. Here, our target is the multi-oriented texts as they present frequently in our daily life.

2.1 Introduction

According to literature, VGG [113] and ResNet [42] are the most widely used backbones in current state-of-the-art text detectors, such as EAST [156], Mask Textspotter [83], SegLink [106], TextBoxes [64], Pixel-Anchor [61], SPCNET [137], etc. As shown in Fig. 2.1, VGG breaks convolutional layers with large receptive fields (e.g., 5×5 or 7×7) into a stack of 3×3 ones to make the network more discriminative and more lightweight, while ResNet utilizes shortcut connections to perform residual learning and ease the gradient vanishing problem. Notably, by introducing shortcut connections, depth of networks has increased remarkably from 19 layers to 152 layers or deeper, leading to a significant performance improvement.

Xception proposed in DeepLab V3+ [14] is developed based on VGG and ResNet. Therefore, apart from the advanced techniques used in VGG and ResNet, Xception also employs other sophisticated techniques to boost its performance, including atrous convolution used to expand the receptive fields without shrinking feature maps' resolution, depthwise separable convolution designed to decrease the number of parameters and ASPP module proposed to combine richer information from multiple levels, as shown in Fig. 2.1. Though Xception-based networks have achieved champion in semantic segmentation, their potential has not been explored in the field of text detection. Motivated by this, in this work, for the first time, we adapt DeepLab V3+ to the task of scene text detection, and improve the detection performance by introducing a series of modifications.

Particularly, in our experiments, we find that the recall rate of our text detector is seriously influenced by the miss detection of small text areas. In DeepLab v3+, an Atrous Spatial Pyramid Pooling (ASPP) module is assembled after the Xception backbone so that richer information can be extracted with various receptive fields. However, features of small text areas have already disappeared from feature maps extracted by the intermediate layers of Xception, so ASPP can do nothing for detecting these text areas. To tackle this problem, in this work, we propose to modify DeepLab v3+ by inserting multiple ASPP modules to it after feature maps with different resolutions so that more detailed and richer information can be extracted for both large text areas and small text areas. Moreover, we also propose to utilize multiple auxiliary Intersection-over-Union (IoU) losses and auxiliary connections to accelerate the training process and enhance the discrimination ability of lower encoder layers.

2.2 Proposed Method

Overview of our proposed DeepText is shown in Fig. 2.2. As we can see, it is an encoder-decoder structure that takes DeepLab v3+ [14] as its base model. The network structure and configuration have been properly modified in this work to adapt DeepLab V3+ to text detection. More details are presented below.

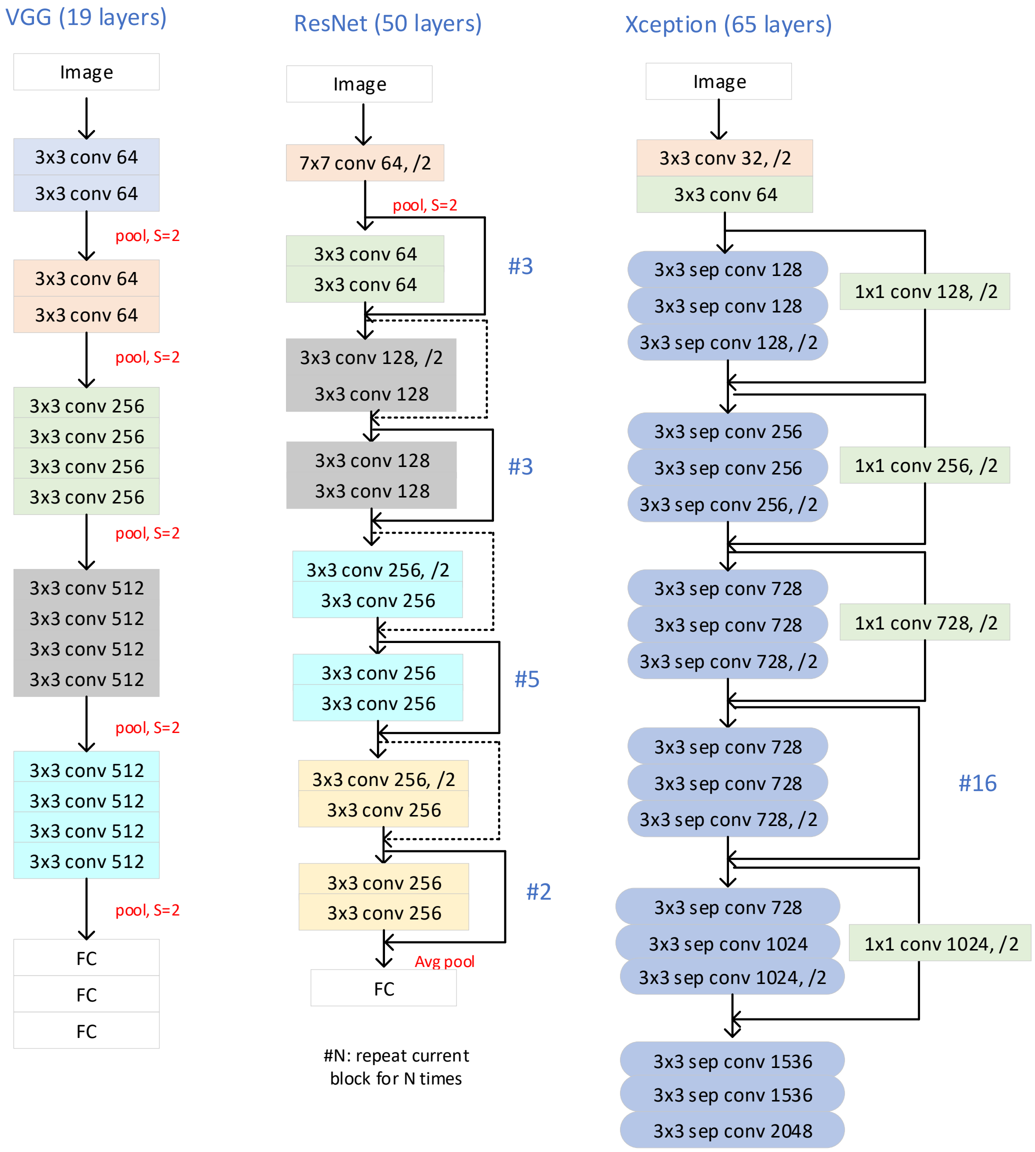


Figure 2.1 : Structure comparison of VGG, ResNet and Xception.

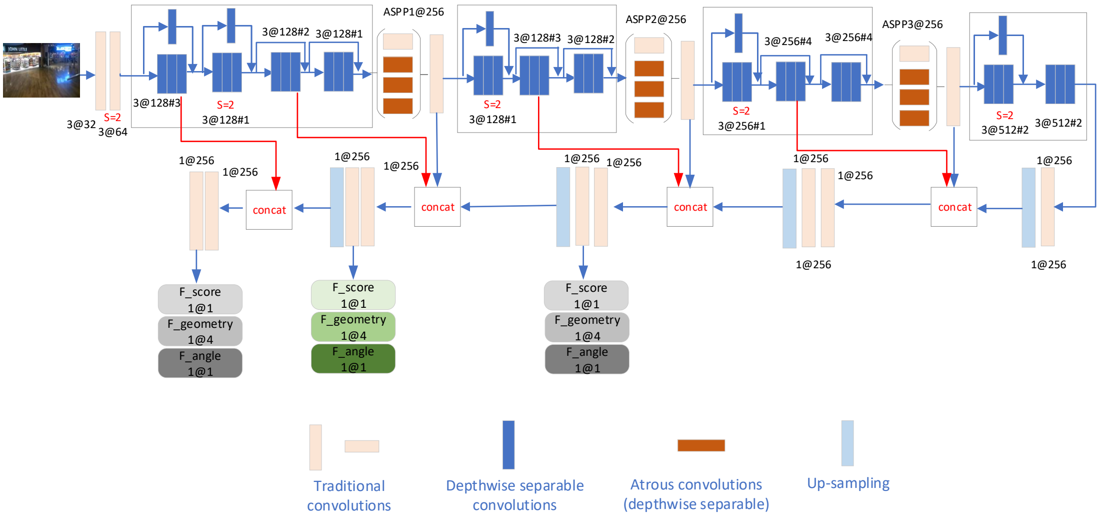


Figure 2.2 : The structure of proposed DeepText network (from [126]), where $a@b\#c$ means current block is with kernel size $a \times a$ and output channel b, and is repeated for c times. $S = 2$ means the stride is set to 2 at a specific layer or the last layer of a specific block.

2.2.1 Backbone of Proposed DeepText

DeepLab v3+ [14] is an efficient semantic segmentation model developed on the base of DeepLab v1 [11], DeepLab v2 [12], and DeepLab v3 [13]. Advanced techniques such as atrous convolution, depthwise separable convolution, ASPP, etc., are exploited in its Xception backbone to generate accurate pixel-level predictions. As shown in Fig. 2.3, atrous convolution adjusts filters' receptive fields and controls feature maps' resolutions by inserting zeros into filters according to pre-defined atrous rates. Note that atrous covolution with rate=1 is equivalent to the standard convolution. ASPP employs multiple atrous convolutional layers in parallel to capture context from multiple scales with different atrous rates, and depthwise separable convolution breaks standard convolution into depthwise convolution and pointwise convolution so that the computation burden can be alleviated. Therefore, to take advantages of these advanced techniques, in this work, we utilize Xception as our backbone.

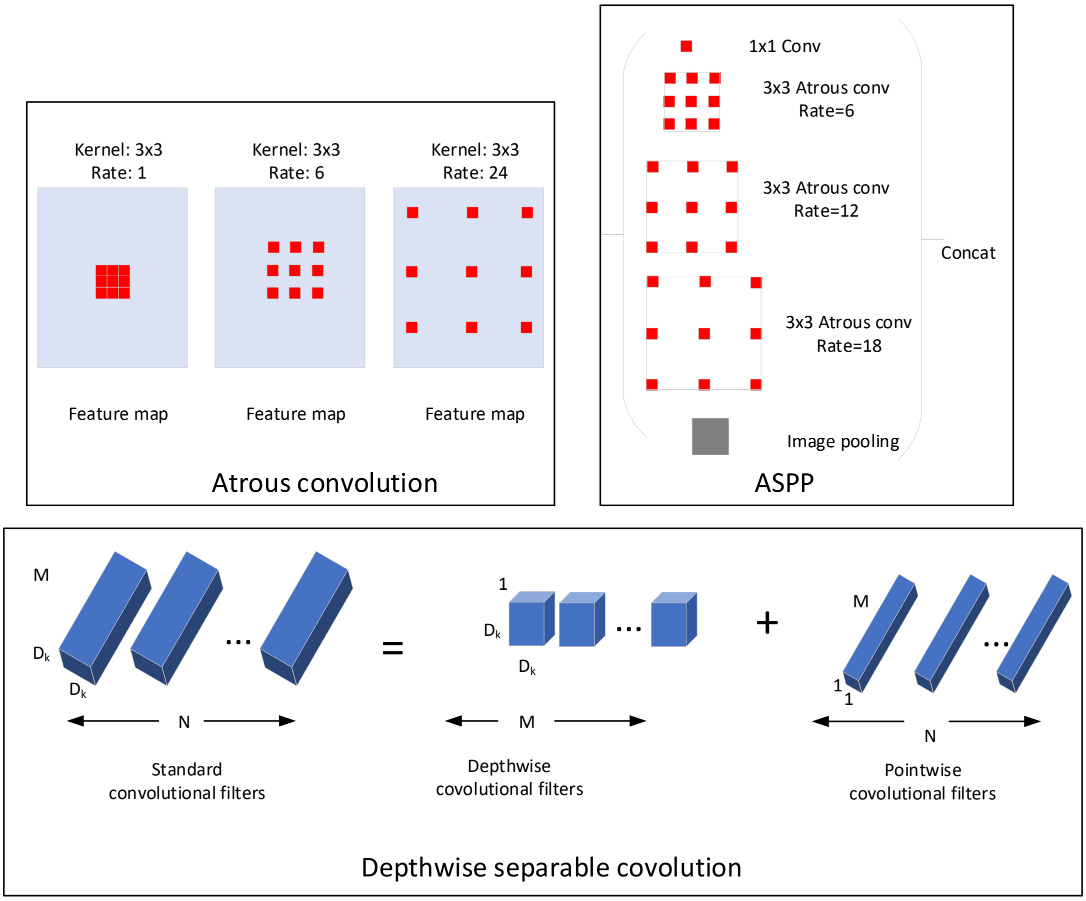


Figure 2.3 : Atrous convolution, ASPP and depthwise separable convolution (from [13, 14]).

Compared with Xception used in [14], we have modified the network structure and configuration as shown in Fig.2.2, where the smallest resolution at the encoder stage is 32, and the recovered resolution at the decoder stage is 2. Moreover, $a@b\#c$ means the current block (each block has 3 depthwise separable convolution layers) is repeated for c times, the kernel size of this block is $a \times a$ and there are b channels at each layer of this block. $S = 2$ means the stride is 2 at a specific layer or the last layer of specific block. Note that, due to the limited GPU resource, denser output feature maps are not considered in our DeepText network, and the channel setting is also much smaller compared to that of [14].

In addition, the features are bilinearly up-sampled with a factor of 16 in the decoder stage of [13], which is claimed [14] to have failed when recovering segmentation details. Therefore, in DeepLab v3+ [14], the up-sampling operation is performed with a factor of 4. In our case, detecting small text areas requires more detailed information and more refined features recovered, so we up-sample feature maps with a smaller factor of 2 at the decoder stage, and then concatenate them with the low-level features from the encoder stage.

2.2.2 Output Layer

The DeepLab v3+ [14] originally proposed for semantic segmentation has a pixel-level prediction module in its output layer, where confidence maps with respect to individual object classes are produced. This prediction layer works well for the semantic segmentation purpose, but is not suitable for our scene text detection task. In order to locate text in images, we also need to predict offsets from individual pixels to the related bounding boxes. Therefore, in this work, we replace the original output layer with a classification module and a regression module.

Concretely, a score map is generated to evaluate pixels' confidence of being text, and five RBOX geometry maps are generated to perform a direct regression, as shown in Fig. 2.4. For an individual location (X, Y), the values at the five RBOX geometry maps represent the distances to the four boundaries of the corresponding rotated box and the rotation angle of the corresponding box, respectively. During the testing stage, we restore the corresponding bounding box according to the prediction results, and eliminate the redundant boxes with the NMS algorithm.

2.2.3 Multiple ASPP Layers

Small texts are present frequently in scene images and detection accuracy of such texts has a great impact on the overall performance. To better deal with these text objects, we further improve the network architecture by inserting multiple ASPP layers to our DeepText network after the feature maps with different resolutions.

The original DeepLab v3+ [14] assembles only one ASPP layer after the feature maps with the smallest resolution (at the end of the encoder). As shown in Fig. 2.5,

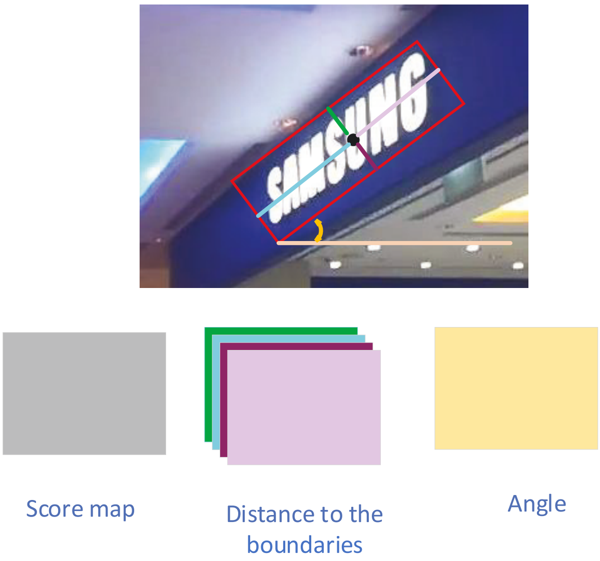


Figure 2.4 : Output feature maps of proposed network (from [126]).

this operation is helpful for extracting wide range contextual information for large texts. However, when it comes to small texts, the extracted features become too coarse, and much detailed information is missed. By contrast, if an ASPP layer with the same atrous rates is applied on the feature maps with a large resolution, the extracted features would be more refined for small texts, but the contextual information contained might be too little for large texts. To take both small texts and large texts into consideration, we propose to insert multiple ASPP layers to the DeepLab after the feature maps with different resolutions. As shown in Fig. 2.2, we assemble three ASPP layers after the feature maps with resolutions of 4, 8, 16, respectively. In an individual ASPP layer, a traditional convolution layer with 1×1 kernel and three atrous convolutional layers with atrous rates of 6, 12 and 18 are assembled in parallel. Then, outputs of these four layers are concatenated, followed by a 1×1 traditional convolutional layer that is used to reduce the overall channels of feature maps.

2.2.4 Multiple Auxiliary Losses and Connections

To optimize our proposed network, the IoU loss [150], as defined in Eq. 2.1, is employed in our work. The IoU loss is originally proposed for object detection.

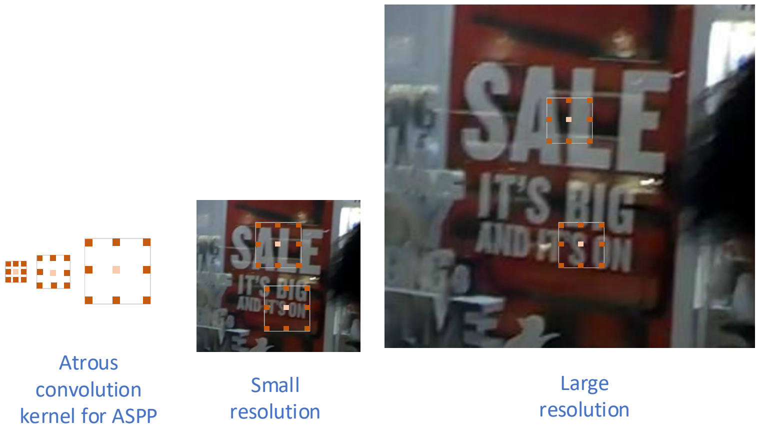


Figure 2.5 : Feature extraction by ASPP with the same atrous rates for text with various scales (from [126]).

Compared with the widely used L2 loss that optimizes the four values of distance independently, the IoU loss is invariant against different scales of objects. Given the predicted bounding boxes R^* and the ground truth bounding boxes R (their related orientations are denoted by θ^* and θ respectively), the IoU loss minimizes the difference between their intersection area and their union area. In our case, the IoU loss is calculated for individual pixels, and the predicted bounding box R^* is derived from the five geometry maps produced by the output layer.

$$\begin{aligned} &Loss_{IoU} = Loss_{area} + Loss_{angle}, \\ &Loss_{area} = -logIoU(R, R^*) = -log\tfrac{|R \cap R^*|}{|R \cup R^*|}, \\ &Loss_{angle} = \lambda * (1 - cos(\theta - \theta^*)). \end{aligned} \tag{2.1}$$

where

$$|R \cup R^*| = |R| + |R^*| - |R \cap R^*|. \tag{2.2}$$

Subsequently, to assist the training of the proposed network and promote the convergence speed, we propose to employ multiple auxiliary IoU losses and connections at the decoder module, which is expected to be able to enhance the gradient signals during the back propagation procedure. The existing scene text detectors usually calculate the loss once on the final decoded feature maps. For example,

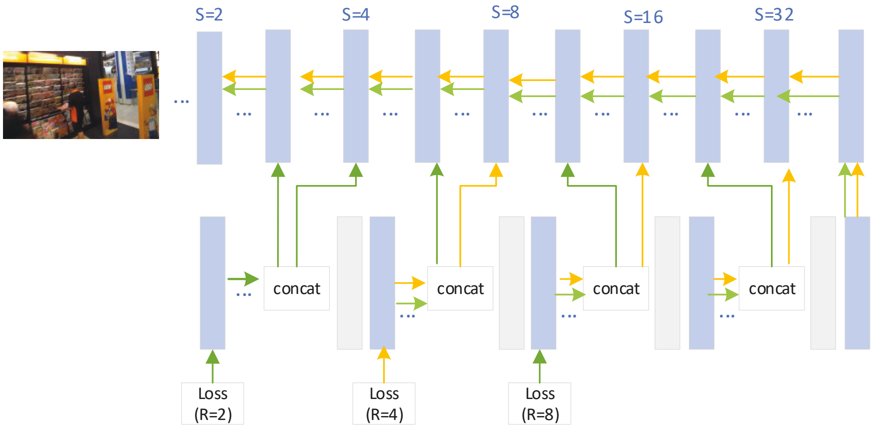


Figure 2.6 : Back propagation after using auxiliary losses and connections. The yellow arrows indicate back propagation paths without auxiliary losses and connections, while the green arrows represent additional paths after using auxiliary losses and connections (from [126]).

EAST [156] calculated the loss on the feature maps with 1/4 resolution and PixelLink [24] did on the feature maps with 1/2 resolution. Moreover, in these models, up-sampled features of the decoder module are often concatenated with the low level feature maps that have the same resolution from only one layer. These strategies make the learning of low level weights slow and the learned features less discriminative. In this work, to enhance the discrimination power of low encoder layers and speed up the convergence, we calculate the IoU loss three times on the feature maps with resolution 1/2, 1/4 and 1/8, respectively, and make auxiliary connections from multiple intermediate encoder layers, as shown in Fig. 2.2. Note that, in the inference stage, we only perform prediction at the feature maps of 1/4 resolution to save time. Fig. 2.6 describes the back propagation details. As we can see, the gradients are enhanced by the auxiliary losses and connections.

2.3 Experiments

To demonstrate the effectiveness of our proposed detector, we test our DeepText on the benchmark dataset ICDAR2015 and compare it with the state-of-the-art approaches.

2.3.1 Datasets

The ICDAR2015 dataset was proposed for the Incidental Scene Text Reading Competition of ICDAR 2015 [52]. Images in this dataset are taken by Google Glasses without limitation on text position, image quality and view point. This dataset is very challenging because text instances could be small, blur and multi-oriented. There are 1000 training images and 500 test images in this dataset, and all of the text regions are labeled with word level quadrangles. We also include 229 training images from the ICDAR2013 dataset in our training set. Therefore, in our experiments, we totally have 1229 training images. Performance of the proposed method is evaluated on the 500 ICDAR2015 test samples.

2.3.2 Implementation Details

To optimize the proposed DeepText network, the Adam optimizer with an initial learning rate of 1e-4 is used. The learning rate is decayed exponentially with a decay rate of 0.94 and a decay step of 10000. The proposed model is implemented with the Tensorflow framework, and our batch size is set to 4 due to the limitation of GPU memory, instead of 8 used in some other literatures.

2.3.3 Evaluation of the Proposed Detector

To demonstrate the effectiveness of our proposed detector, we compare the performance with those of state-of-the-art approaches. Table 2.1 gives details of the comparison results.

As we all know, training data has a great impact on detection performance, so we include additional training sets when other training samples in the datasets are used in addition to the ICDAR2013 and ICDAR2015 datasets. The results tested on multiple scales can always be better than those tested on a single scale. Since many methods only report their results on a single scale, to be fair, we only list single scale results for all of the methods in Table 2.1. When multiple settings are tested for certain models, we report their best ones. For example, the model named EAST tests seven settings in [156], but we only take their best performance achieved by

Table 2.1 : Comparison with prior arts on ICDAR2015 (from [126]).

Methods	**Additional data**	**Recall**	**Precision**	**F-measure**
EAST [156]	ImageNet	73.47	83.57	78.20
SegLink [106]	SynthText	76.80	73.10	75.00
RRPN [85]	ImageNet, SVT	73.23	82.17	77.44
R^{2}CNN [49]	ImageNet	74.29	76.42	75.34
TextBoxes++ [63]	SynthText	76.70	87.20	81.70
PixelLink [24]	No	82.00	85.50	83.70
TextSpotter [83]	SynthText	81.20	85.80	83.40
DeepLab_small	No	77.03	86.21	81.36
DeepLab	No	77.80	87.49	82.36
DeepLab_MASPP	No	81.08	87.57	84.20
Proposed DeepText	No	81.13	88.27	84.55

PVANET2x on a single scale. Additionally, if the compared method is an end-to-end method, we take their detection-branch-only results in Table 2.1, such as Mask TextSpotter. ICDAR 2015 does not provide any offline evaluation tool or ground truth for the test set. Therefore, we directly submit our prediction results to the online platform (http://rrc.cvc.uab.es/?ch=4&com=evaluation&task=1) and take the platform's evaluation results.

From Table 2.1, we can see that the proposed DeepText achieves the best performance among all of the listed detectors with a F-measure of 84.55%. Notably, all of the listed detectors pre-train their models using additional datasets such as ImageNet, SynthText, etc., except for PixelLink and ours. To demonstrate the effectiveness of our modification, we also carry out experiments with the original DeepLapv3+ [14] structure, indicated by DeepLab in Table 2.1. DeepLab_small has the same structure and layer setting as DeepLab, but the channels in each layer is shrunk from 256 to 128 (layers with 256 channels in Fig. 2.2) and from 512 to 256 (layers with 512 channels in Fig. 2.2). DeepLab and DeepLab_small use the same loss function, data pre-processing strategies, learning rate and optimizer as EAST. The only difference is that EAST uses VGG as the backbone. Clearly, DeepLabv3+ is a better backbone than VGG because the performance is elevated from 78.20% to 82.36% (for DeepLab). Even we use a smaller setting for

DeepLab_small, the F-measure is 3.16% higher. Furthermore, the performance of DeepLab_small is 1% lower than that of DeepLab, so we can conclude that greater setting is good for the improvement of model's performance. Therefore, when comparing the performances of different models, both of the structure and the network scale should be taken into consideration. Unfortunately, due to the limitation of our GPU memory, we cannot implement our model with a bigger setting and compare the performance with the methods like IncepText, which has 1024, 2048 and 1024 channels in convolution stage-4, convolution stage-5 and the decoder stage, respectively, and achieves a performance of 85.3% when a single scale is used.

The method named as DeepLab_MASPP in Table 2.1 has the same settings (number of layers and channels in each layer) as the one named as DeepLab, except that DeepLab_MASPP utilizes multiple ASPP layers in the encoder stage and up-samples feature maps with a factor of 2 at the decoder stage. Apparently, when MASPP and smaller up-sample factors are used, the performance can be significantly improved because more smaller text regions are recalled (the recall is improved from 77.80% to 81.08%). Finally, after employing multiple auxiliary IoU losses and auxiliary connections, we obtain a detection performance of 84.55%, which is slight better than that of DeepLab_MASPP. However, DeepLab_MASPP gets the best results of 84.20% at the iteration of 1154k (with the batch size set to 4), while after using auxiliary losses and connections, the best results of 84.55% is obtained at the iteration of 734k (with the batch size also set to 4). It is evidenced that auxiliary losses and connections are able to greatly assist the training of deep networks in the text detection task, and the discrimination of lower encoding layers can also be enhanced.

2.4 Conclusion

A powerful backbone is essential to deep networks in the field of computer vision. In this paper, we have firstly introduced the well-known DeepLab structure for the scene text detection task, and achieved promising performance. When detecting text from scene images, encoding the wider range contextual information and detailed information from different scales is able to improve models’ robustness to arbitrary text sizes and orientations. Toward this end, we have modified the original DeepLab structure by inserting multiple ASPP layers to the network after feature maps with different resolutions. Additionally, multiple auxiliary IoU losses and connections have been employed to assist the network training and enhance the discrimination ability of lower encoder layers. Experimental results on ICDAR2015 have shown that the performance has been significantly improved by applying the proposed modifications.

Chapter 3

Mask R-CNN with Global Text Context for Multi-lingual Multi-oriented Text Detection

It is a common sense that in modern cities, multiple cultures live and communicate together. Therefore, text detection is not only struggling with the arbitrary orientation problem but also facing challenges posed by multiple scripts and languages. On the other hand, as previously introduced, texts can be regarded as a kind of particular objects in scene images, so it is a natural thought to detect text with general object detection framework. Given above observations, in this chapter, we propose a multi-lingual multi-oriented text detector on the base of Mask R-CNN [40], which is a well-known two-stage object detector.

3.1 Introduction

According to literature, general object detectors can be classified into two subgroups, *i.e.,* one-stage detectors (e.g., SSD [73] and YOLO [97]) and two-stage detectors (*e.g.,* Fast/Faster R-CNN [98, 33] and Mask R-CNN [40]). The former have the advantage of speed since they only perform classification and regression once for individual default boxes, while the latter achieve higher detection accuracies because default boxes are classified and refined twice in a cascade way. For an intuitive comparison of above two frameworks, we presents structures of SSD and Faster R-CNN in Fig. 3.1, where $Conv\ 3 \times 3 \times (M \times (C + 4))$ means a 3×3 convolutional layer with $M \times (classes + 4)$ output channels is used for classification and regression. Here, M, C and 4 denote the number of default boxes at each location, the total number of object categories and the number of regression offsets for each default box, respectively.

As mentioned previously, many state-of-the-art text detectors follow the prac-

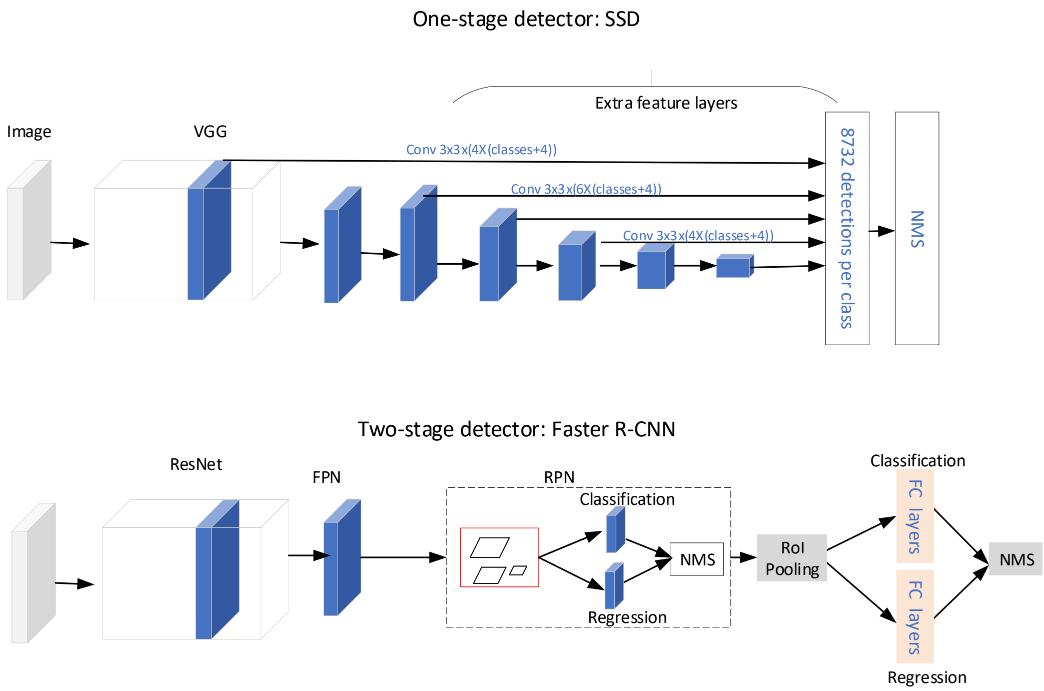


Figure 3.1 : Overview of one-stage object detector and two-stage object detector.

tice in general object detection. They can be motivated by either one-stage detectors [43, 64, 63, 106, 66, 100] or two-stage detectors [77, 85, 152, 121, 23, 158]. Currently, those based on Mask R-CNN [71, 46, 70] achieve the highest performances on the challenging multi-lingual multi-oriented text dataset, *i.e.,* MLT. Therefore, in this work, we follow the same trend and design our text detector based on Mask R-CNN. Mask R-CNN is an enhanced version of Faster R-CNN, so it is developed under the two-stage framework. But, unlike Faster R-CNN, Mask R-CNN utilizes RoIAlign to extract feature vectors from arbitrary-size text proposals, instead of RoIPool. Moreover, in the second classification and regression stage, an additional mask prediction branch is assembled in Mask R-CNN to perform pixel-level predictions inside preserved text proposals so that contours and minimum bounding rectangle of target text areas can be inferred.

Though detectors derived from Mask R-CNN have achieved champion in text detection, their performance is still limited by the data augmentation strategies.

Following the standard data augmentation routine in general object detection, horizontal flipping, randomly resizing and normalization are performed on input images in [137, 46]. However, as proved in [70], image cropping used in other tasks also works well for text detection. Therefore, in this work, we follow [70] to design our data augmentation strategies.

RoIAlign plays a critical role in Mask R-CNN. It extracts fixed-size feature maps from arbitrary-size text proposals so that network's efficiency can be retained when dealing with thousands of text proposals. However, as pointed out in [137], RoIAlign extracts information from only local RoI regions, so it suffers from false positives and inaccurate classification scores caused by the lack of context information clues. To address this issue, Xie et al. [137] proposed a Text Context Module (TCM) and a re-score mechanism in their SPCNET, but, unfortunately, their model's efficiency seems to be compromised significantly because of the newly added modules. To efficiently make use of the global text context, in this work, we propose to conduct semantic segmentation from a global view on feature maps generated by FPN, and guide the subsequent classification, regression and mask prediction modules with the segmentation results. Details of our proposed GMask R-CNN, *i.e.,* Mask R-CNN with Global Text Context, can be found below.

3.2 Proposed Method

Structure of our proposed GMask R-CNN is depicted in Fig. 3.2. As we can see, it takes Mask R-CNN as its base model and is equipped with a newly designed global mask module, as indicated in the red dotted rectangle. Besides, feature maps produced by FPN are enhanced by the predicted text masks before being fed into the subsequent two-stage classification and regression modules. According to our experiments, the speed of the calculation is not affected obviously since only three layers are added for each scale of the feature pyramid.

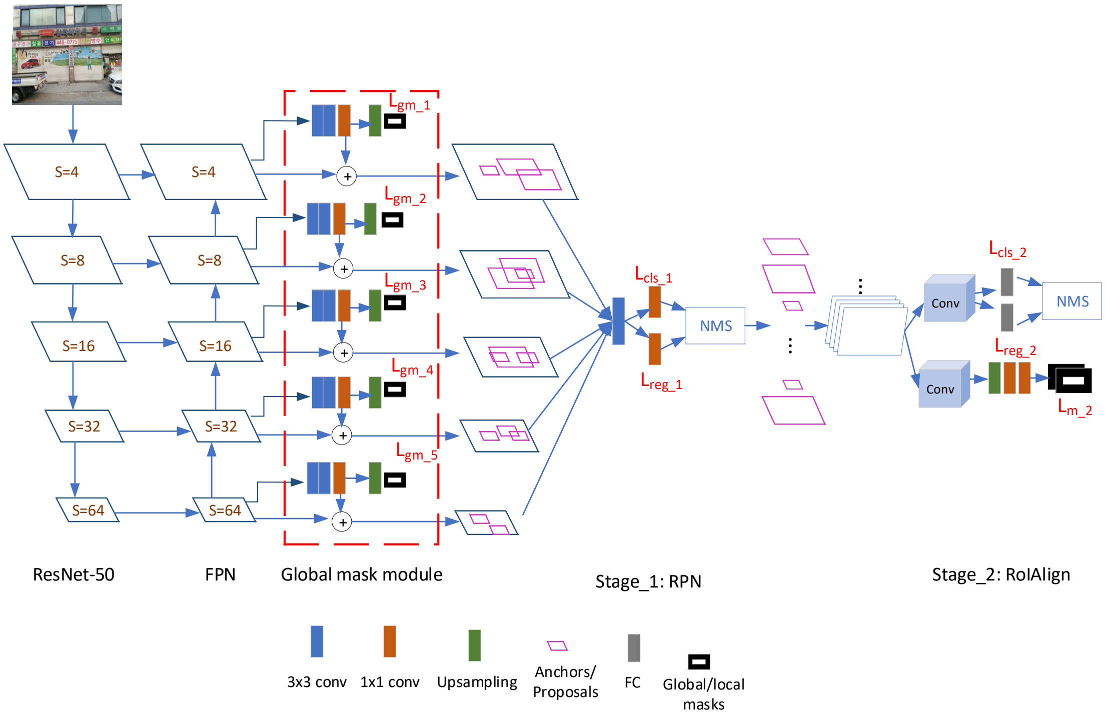


Figure 3.2 : Network structure of proposed GMask R-CNN.

3.2.1 Mask R-CNN

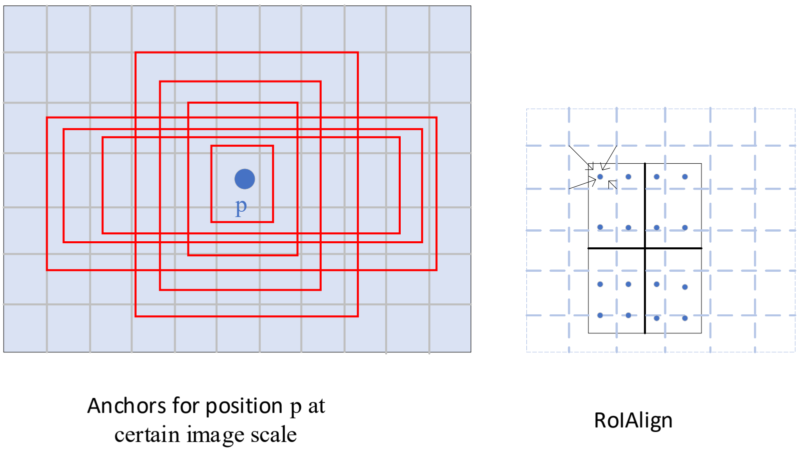
As illustrated in Fig. 3.2, Mask R-CNN employs ResNet to extract features from input images, followed by a top-down architecture named FPN [67], which is utilized to fuse feature maps from multiple scales. Then, according to the pre-defined parameters, *i.e.*, number of scales, aspect ratios, base sizes, number of sampling points, etc., millions of anchors are produced, refined and filtered out by the subsequent RPN module. Fig 3.3 presents anchors generated for position p at image scale l. Assuming there are k anchors for each position, RPN exploits two parallel convolutional branches with $2K$ (for text/non-text scores) and $4K$ (4 predicted offsets for each anchor) output channels to perform classification and regression, respectively. Afterwards, NMS is set at the end of RPN to select thousands of refined anchors with considering their text/non-text classification scores. The selected anchors are with the highest probabilities to be texts and thus are called text proposals in the second stage.

At the beginning of the second stage, RoIAlign proposed in [40] is exploited to extract fixed-size feature maps from arbitrary-size text proposals. As illustrated in Fig 3.3, each text proposal is evenly divided into M bins, and each bin has N sampling points. RoIAlign calculates values for individual sampling points from their nearby grid points via the widely used bilinear interpolation algorithm. Then, after being further encoded, the extracted feature maps are converted into feature vectors and fed into two parallel fully connected layers for the second-stage regression and classification. Finally, according to the prediction results, text proposals are further refined and selected by NMS to generate expected output detections. Note that a mask generation branch is also designed in the second stage to perform pixel-level predictions inside individual text proposals so that contours and minimum bounding rectangular of our targets, *i.e.,* multi-oriented texts, can be inferred.

In the mask prediction branch, Mask R-CNN exploits a deconvolution layer to upsample feature maps. As claimed in [71], deconvolution suffers from the checkerboard problem, which is harmful to the following pixel-level prediction. Therefore, inspired by [71], we replace the deconvolution layer with a bilinear interpolation layer and a 1×1 convolutional layer.

3.2.2 Global Mask Module

The newly added global mask module is constructed by multiple groups of 3×3 convolutional layers, 1×1 convolutional layer and bilinear up-sampling layer. At each scale of the feature pyramid, the 1×1 convolutional layer is firstly employed to produce a single channel text mask with the same size as feature maps of current scale. Then, to calculate the global mask prediction loss, we employ an up-sampling layer to rescale the predicted text mask to the same size as input images. In addition, at individual scales, we concatenate the predicted text masks, i.e., outputs of the 1×1 convolutional layers, with feature maps produced by FPN so that the subsequent second-stage classification and regression modules can take advantage of the global text context.



Anchors for position p at certain image scale

RoIAlign

Figure 3.3 : Default anchors generated for position p at image scale l and RoIAlign for text proposals.

Benefits of the proposed global mask module are twofold. Firstly, by introducing additional global mask prediction losses, optimization of previous ResNet-50 and FPN will be boosted. Therefore, feature maps generated by FPN will be more discriminative. Secondly, these global masks are predicted with considering contextual information, so the classification and regression accuracies of anchors and text proposals can be improved by enhancing feature maps of FPN with the predicted global masks. Experimental results presented below will demonstrate the effectiveness of our proposed global mask module.

3.2.3 Loss Function of Proposed GMask R-CNN

Loss function of the proposed GMask R-CNN consists of multiple components, as formulated in Eq. 3.1, where L_{gm}, L_{rpn} and L_{RoI} denote global mask prediction loss, first stage RPN loss and second stage RoI prediction loss, respectively. As indicated in Fig. 3.2, L_{gm}, L_{rpn} and L_{RoI} can be calculated with the way shown in Eq. 3.2, where N is the number of FPN scales and $\lambda_1, \lambda_2, \cdots, \lambda_6$ are coefficients used to balance individual losses. Here, we compute the average mask prediction loss of individual scales for the newly added global mask module. In addition, in this work, we calculate L_{gm_i}, L_{cls_1} and L_{m_2} with the binary cross entropy loss,

L_{cls_2} with the cross entropy loss, and L_{reg_1} as well as L_{reg_2} with the smooth L_1 loss.

$$Loss = L_{gm} + L_{rpn} + L_{RoI}. \tag{3.1}$$

$$\begin{aligned} L_{gm} &= \tfrac{\lambda_1}{N} * \textstyle\sum_{i=1}^{N} L_{gm\_i}, \\ L_{rpn} &= \lambda_2 * L_{cls\_1} + \lambda_3 * L_{reg\_1}, \\ L_{RoI} &= \lambda_4 * L_{cls\_2} + \lambda_5 * L_{reg\_2} + \lambda_6 * L_{m\_2}. \end{aligned} \tag{3.2}$$

3.2.4 Data Augmentation and Configurations

In our experiments, we have conducted some data augmentation operations to improve the detection performance. Firstly, we randomly flip training images in the horizontal direction with a probability of 0.5, and randomly resize height and width of input images to a range of $[768, 2560]$, without keeping their aspect ratios. Then, image pixels are normalized to [0, 1] with the mean value and standard deviation value learned from the ImageNet dataset. Besides, we also randomly crop training images to a size of 768×768 since the cropping operation is able to improve the detection performance significantly. By contrast, in general object detection, Mask R-CNN resizes training images to a minimal size of 800 and maximal size of 1333 while keeping their aspect ratios and does not adopt the cropping operation. Note that in the prediction stage, we resize test images to a maximal size of 1600 or 1920 while keeping their aspect ratios, and all the test results are evaluated on the single scale.

On the other hand, though texts can be regarded as a kind of particular objects, they still have some unique characteristics such as larger aspect ratios, longer text lengths, etc. Therefore, to adapt GMask R-CNN to text detection, we change the base sizes and aspect ratios of anchors to $(16, 32, 64, 128, 256)$ and $(0.17, 0.44, 1.13, 2.90, 7.46)$, respectively. Moreover, Mask R-CNN and PMTD [71] are claimed to have achieved their best performance at a small score threshold of 0.05 in the NMS module of the second stage. However, in our experiments, we find that a score threshold of 0.85 is more appropriate.

3.3 Experiments

The proposed text detector has achieved promising performance on IC15 and MLT datasets, details and comparison results are presented in this section.

3.3.1 Datasets

We evaluate performance of our proposed GMask R-CNN on the ICDAR 2015 and ICDAR 2017 robust text reading competition datasets, *i.e.,* IC15 and MLT. IC15 is an English multi-oriented dataset, which has been used in our work previously, and MLT is a multi-lingual multi-oriented dataset, where 9 languages are involved, including English, Chinese, Japanese, Korean, etc. MLT consists of 7200 training images, 1800 validation images and 9000 test images. Following the practice in other works, we employ all of the training and validation images to optimize our network and evaluate the performance on the test set.

3.3.2 Implementation Details

We pre-train the ResNet-50 backbone on ImageNet [25], a large-scale image classification dataset, and fine-tune the whole network on IC15 and MLT. In particular, the pre-trained network is firstly trained on MLT for 168 epochs with the widely used SGD optimizer. Our experiments are conducted on two 16G GPUs, so the batch size is set to 10. Accordingly, the learning rate is warmed up exponentially from 0.00125 to 0.0125 in the first 5000 iterations and decays to 0.00125 and 0.000125 at the 88th epoch and 128th epoch, respectively. The training procedure on MLT is terminated at 168th epoch. Afterwards, the network is continually fine-tuned on IC15 for another 40 epochs with a fixed learning rate of 0.000125.

3.3.3 Comparison Results

We compare performance of our proposed GMask R-CNN with that of the existing state-of-the-art text detectors in Table 3.1 and Table 3.2. Note that on both MLT and IC15, the performance of the proposed detector is evaluated on

Table 3.1 : Comparison with state-of-the-art approaches on MLT dataset. Our proposed GMask R-CNN is tested with size 1600 and 1920, indicated by *_1600 and *_1920, respectively.

Methods	**Recall**	**Precision**	**F-measure**
FOTS [76]	57.51	80.95	67.25
Lyu et al. [84]	55.60	83.80	66.80
PSENet [60]	68.40	77.01	72.45
Pixel-Anchor [61]	59.54	79.54	68.10
SPCNET [137]	73.40	66.90	70.00
Huang et al. [46]	69.80	80.00	74.30
Baseline_1600	69.21	84.00	75.89
Proposed GMask R-CNN_1600	71.45	82.32	**76.50**
Proposed GMask R-CNN_1920	73.48	81.50	**77.29**

Table 3.2 : Comparison with state-of-the-art approaches on ICDAR2015.

Methods	**Recall**	**Precision**	**F-measure**
EAST [156]	73.47	83.57	78.20
SegLink [106]	76.80	73.10	75.00
RRPN [85]	73.23	82.17	77.44
R^{2}CNN [49]	74.29	76.42	75.34
TextBoxes++ [63]	76.70	87.20	81.70
PixelLink [24]	82.00	85.50	83.70
TextSpotter [83]	81.20	85.80	83.40
DeepText [126]	81.13	88.27	84.55
FTSN [76]	80.00	88.60	84.10
TextSnake [82]	80.40	84.90	82.60
SPCNET [137]	85.80	88.70	87.20
PSENet [60]	85.22	89.30	87.21
FOTS [76]	85.17	91.00	87.99
Proposed GMask R-CNN	87.48	90.31	**88.87**

single scale, where the long side of input images is resized to 1600 or 1920, indicated by $*$_1600 and $*$_1920, respectively.

As shown in Table 3.1 and Table 3.2, our proposed detector outperforms the existing state-of-the-art approaches on both MLT and IC15 datasets. We attribute most of this success to the powerful Mask R-CNN-based model and the effective data augmentation strategies, especially the cropping operation. Among the listed

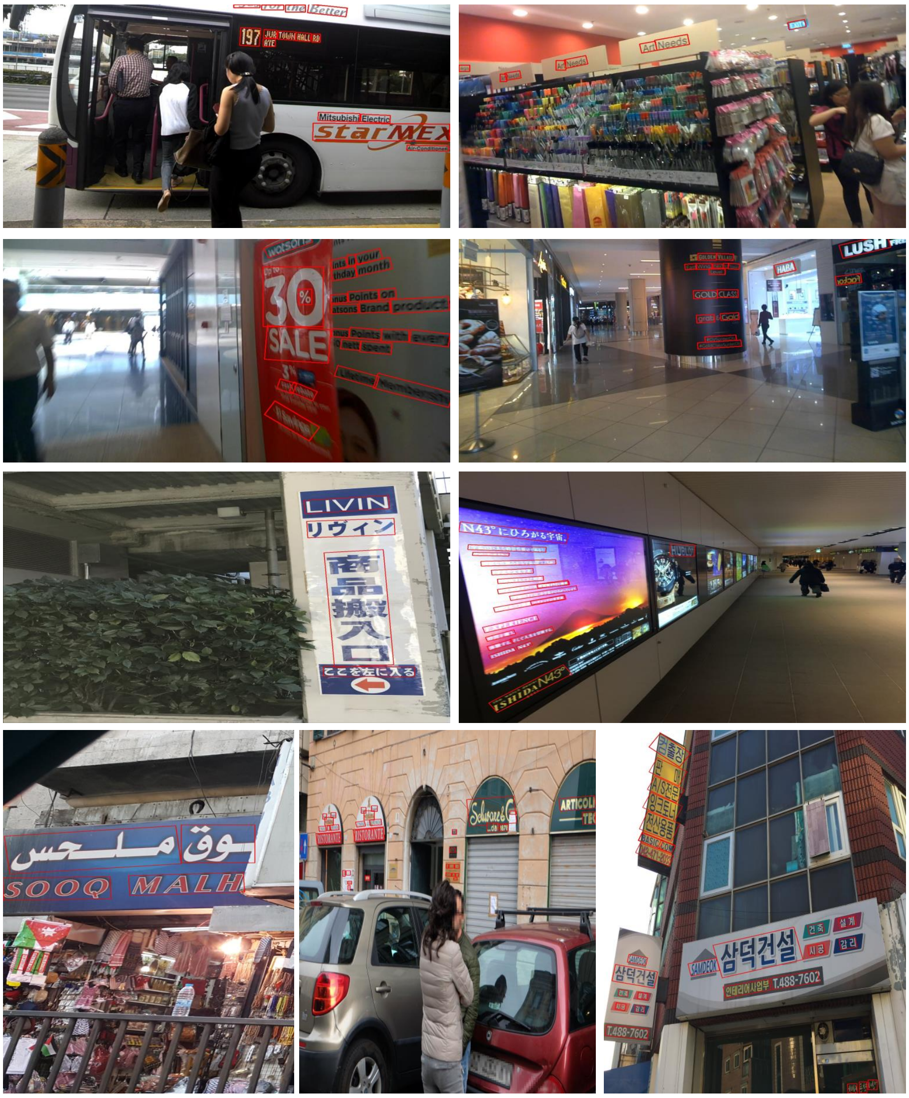


Figure 3.4 : Qualitative results of proposed detector.

methods, SPCNET [137] is also on the base of Mask R-CNN, but its baseline model only achieves an F-measure of 65.50%, which is about 10% lower than ours because of the differences in data augmentation strategies. In addition, from the comparison of the baseline model and our proposed GMask R-CNN, when the proposed global mask module is embedded, performance of our detector is further elevated from 75.89% to 76.50%, and according to our experiments, the value of F-measure also becomes much more stable as the score threshold used before NMS varying. Experimental results also show that when a larger image size of 1920 is used for evaluation, F-measure of our detector can be further improved to 77.29%, so there is a trade off between accuracy and efficiency during the evaluation.

To intuitively illustrate detection performance of our detector, we have shown some qualitative results in Fig. 3.4. As we can see, the proposed text detector work well in very complicated scenarios and is robust to languages and text directions.

3.4 Conclusion

Text can be regarded as a kind of specific objects presenting in scene images, so it is a natural thought to handle text detection from the perspective of object detection. In this work, we have adopted Mask R-CNN, one of the most well-known object detector, as our base model, and have combined it with a newly designed global mask module. Experimental results on MLT and IC15 have demonstrated that the proposed network works well in handling multi-lingual multi-oriented scene text detection, and proper data augmentation strategies play a key role in this procedure. Moreover, ablation study has shown that the proposed global mask module is able to improve detection performance to some extent.

Chapter 4

ConvLSTM-based Neural Network for Scene Text Recognition

Scene text recognition, *i.e.,* transcribing detected text areas into human readable ASCII characters, is a subsequent step of scene text detection in text reading systems. It has received considerable attentions from the community of computer vision and document analysis in past decades. However, because of the challenges posed by poor image qualities (*e.g.*, low resolution, blur, uneven illumination, etc.) and various text appearances (*e.g.*, size, fonts, colours, directions, perspective view, complex background, etc.), as shown in Fig. 4.1, though many efforts have been made, scene text recognition is still an unsolved and challenging task. Therefore, in this chapter, we explore how to recognize texts from scene images by proposing a novel recogniser named FACLSTM [127].

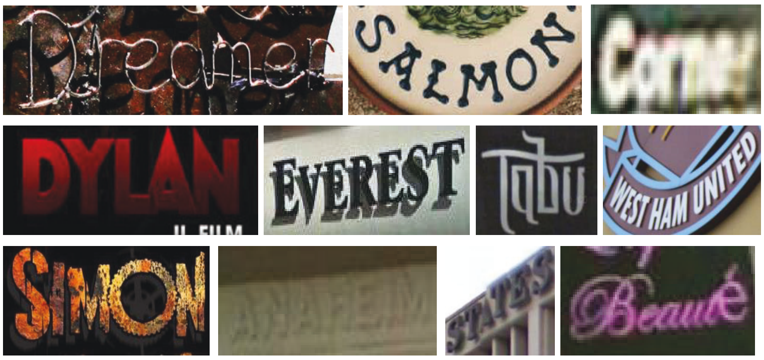


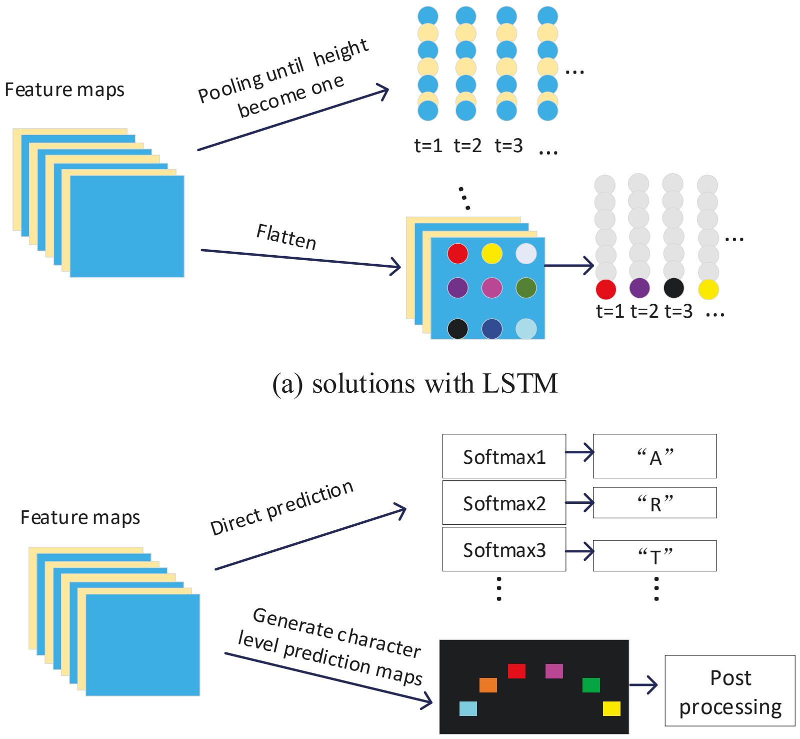
Figure 4.1 : Challenging samples of scene text recognition (from [127]).

4.1 Introduction

Inspired by speech recognition and machine translation, most of recent state-of-the-art approaches regard scene text recognition as a sequence-to-sequence prediction problem and widely adopt techniques like LSTM [45] and attention mechanism [3] in their sequential transcription module. However, LSTM used in these recognisers is the fully-connected-LSTM (FC-LSTM) that only takes stream signals like sentences or audio as inputs and connects them in a fully connected way, while scene text recognition generates sequential outputs from 2-D images. To adapt FC-LSTM to scene text recognition, the most straightforward way is pooling 2-D feature maps to a height of one or flattening them into 1-D sequential feature vectors [32, 15, 17, 107, 108], as shown in Fig. 4.2(a). Unfortunately, such operations could severely disrupt the valuable spatial correlation relationships among pixels, which is essential to computer vision tasks, especially to scene text recognition, where the structures of strokes are the key factors to discriminate characters. To retain such important spatial and structural information, researchers have also explored other alternative solutions. For example, STN-OCR [7] directly performed sequential prediction on 2-D feature maps with a fixed number of softmax classifiers, and CA-FCN [65] generated character-level confidence maps with a fully convolutional network, as shown in Fig. 4.2(b). However, compared with LSTM, these solutions often introduce additional parameters or post processing steps.

In this work, we argue that scene text recognition is essentially a spatio-temporal prediction problem for its 2-D image inputs, and propose a convolution LSTM (ConvLSTM)-based scene text recogniser, namely, FACLSTM, *i.e.,* Focused Attention ConvLSTM, where the spatial correlation of pixels is fully leveraged when performing sequential prediction with LSTM. ConvLSTM is proposed by Shi et al. [112] for precipitation nowcasting. In ConvLSTM, all of the fully connected operations are replaced by the convolutional ones, so input feature maps are allowed to keep their 2-D shapes when being fed into the ConvLSTM-based modules. Given this advantage, for the first time, we introduce ConvLSTM to scene text recognition and apply it in the sequential transcription module of our proposed recogniser.

On the other hand, in the existing models, LSTM is only used for frame-level prediction and is incapable of producing sequential outputs from one single input image unless the CTC or attention mechanism is incorporated. To perform



(a) solutions with LSTM

(b) solutions without LSTM

Figure 4.2 : Current solutions for scene text recognition (from [127]). When using LSTM, 2-D feature maps are usually converted to 1-D space by pooling or flattening operations. When the LSTM is not used, additional parameters or post-processing steps are involved.

sequential prediction and, meanwhile, provide the model spatial awareness, we further improve ConvLSTM by embedding the attention mechanism into the structure. Notably, different from the existing attention-LSTM-based recognisers, where the attention mechanism and FC-LSTM are combined in a fully connected way, we properly integrate the attention mechanism into ConvLSTM with the convolutional operations. Moreover, as ConvLSTM extends 2-D operations into 3-D, the costs of computation and memory increase significantly. To achieve high efficiency, inspired by Liu et al. [72], we propose to assemble a bottleneck gate at the beginning of the proposed attention-equipped ConvLSTM, so that the internal feature map channels can be reduced.

Last but not the least, since the existing attention-based recognisers often suffer from the 'attention drift' problem [15], *i.e.*, they fail to align target outputs to proper feature areas, we propose to learn additional character centre masks with a second decoder branch in the encoder-decoder feature extraction stage to assist the proposed network to focus attention on right feature areas.

The experimental results conducted on benchmark datasets demonstrate that our proposed recogniser is able to achieve comparable performance with the state-of-the-art approaches on regular, low-resolution and noisy text and outperforms other methods significantly on the more challenging curved text.

The contributions made in this work are summarized as follows. (1) We propose to handle the scene text recognition problem from a spatio-temporal prediction perspective and for the first time introduce ConvLSTM to this application. (2) We design a ConvLSTM-based sequential transcription module, where the attention mechanism is harmoniously embedded into ConvLSTM with convolutional operations, and the bottleneck gate is assembled at the beginning of ConvLSTM to retain its efficiency. (3) We propose to learn additional character centre masks to help the proposed network focus attention on the centre of characters.

4.2 Proposed Method

In this work, aiming to better consider the spatial and structural information of input images when performing sequential prediction with LSTM, for the first time, we propose an attention-equipped ConvLSTM structure in the sequential transcription module, and further design a focused attention module to help learn more accurate alignment between predicted characters and corresponding feature areas.

As illustrated in Fig. 4.3, our proposed FACLSTM, *i.e.,* Focused Attention ConvLSTM, consists of two components, *i.e.*, the CNN-based feature extraction module and the ConvLSTM-based sequential transcription module. The feature extraction module is an encoder-decoder structure that takes VGG-16 as the backbone, while the sequential transcription module is a combination of ConvLSTM

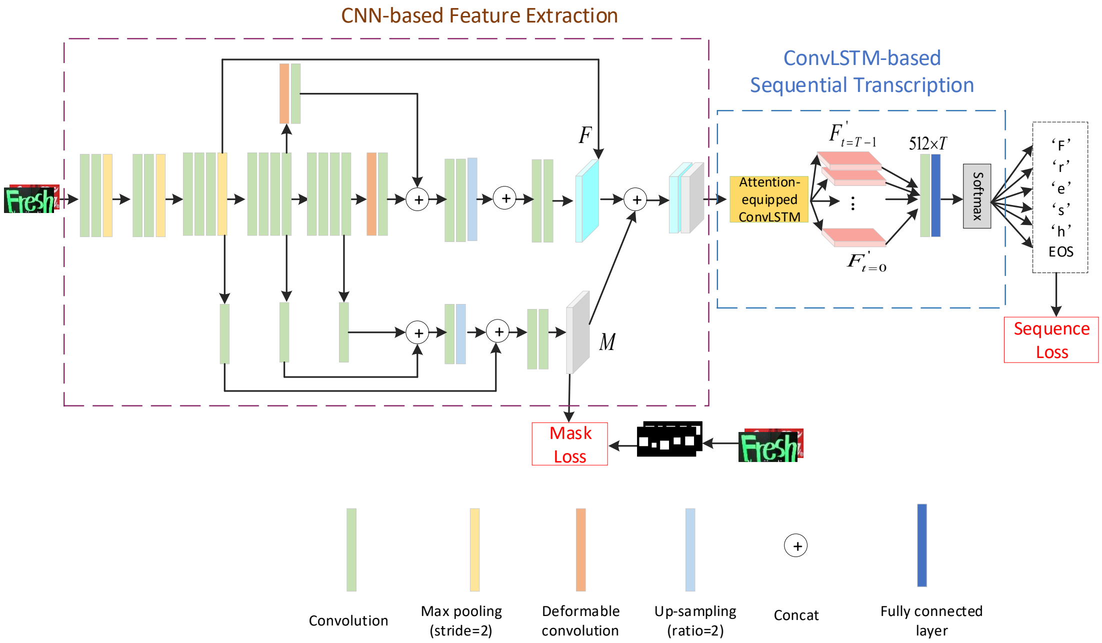


Figure 4.3 : Overview of proposed FACLSTM (from [127]). F and M denote the extracted feature maps and character centre masks. T groups of feature maps are produced by the proposed attention-equipped ConvLSTM, where T is the maximal string length, and the followed softmax classifier is responsible for producing T groups of feature maps from extracted feature maps. Note that, the softmax classifier and previous fully connected layer are shared by the T groups of feature maps.

and attention mechanism. More details are presented as follows.

4.2.1 CNN-based Feature Extraction

Backbone: Similar to Liao's work [65], we take VGG-16 as the encoder of our feature extraction module, and remove the fully connected layers and pooling layers from the last two encoding stages. We also assemble two deformable convolutional layers [22] at stage-4 and stage-5 of the decoder given their flexible receptive fields. However, compared with Liao's network [65], the resolution of final feature maps is restored to a smaller size of $\frac{W}{4} \times \frac{H}{4} \times C$ in our FACLSTM, instead of the $\frac{W}{2} \times \frac{H}{2} \times C$ used in [65], considering the memory and computation cost. Here, W, H

and C denote the width, height and channels of feature maps, respectively. In addition, we remove their character attention module set in the encoder stage, and meanwhile, design a focused attention module in the higher-level decoder stage so that more abstract and powerful character centre masks can be extracted. The extracted masks are sent to the followed sequential transcription module together with features produced by another decoder branch. By contrast, character centre masks produced in [65] are used to enhance features generated at the encoder stage.

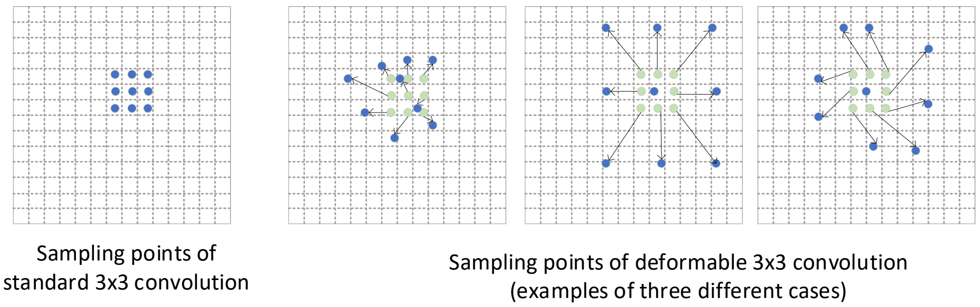
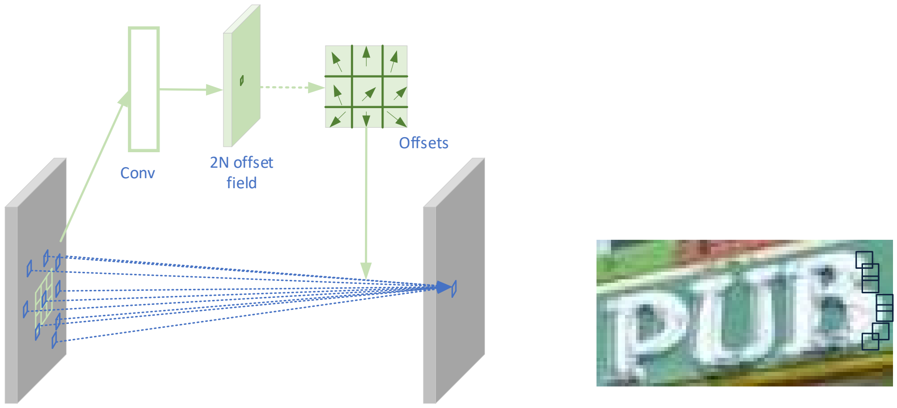

Sampling points of standard 3x3 convolution
Sampling points of deformable 3x3 convolution (examples of three different cases)

Figure 4.4 : Sampling points in standard convolution and deformable convolution. Blue points are the sampling points and arrows indicate offsets of sampling locations.

Deformable Convolution: Objects including characters in texts are usually with arbitrary geometric shapes. However, standard convolution employs fixed geometric structures, resulting in a limited geometric transformation ability. Dai et al. [22] proposed deformable convolution to learn offsets for sampling points so that tasks with unknown geometric transformations could be handled. As shown in Fig. 4.4, deformable convolution can deal with various transformations in terms of scale, rotation, etc., and its offsets of sampled locations are automatically learned according to specific tasks with the way shown in Fig. 4.5. In our application, as clarified in [65], the 3×1 deformable convolution is used for more precise boundary prediction between texts and backgrounds because of its transformable and flexible receptive fields.

Focused Attention Module: As pointed out in [15], current attention-based



Calculation of deformable convolution

Usage of 3x1 deformable convolution in text recognition

Figure 4.5 : Illustration of deformable convolution.

models suffer from the 'attention drift' problem, *i.e.*, they fail to obtain an accurate alignment between target characters and related feature areas, especially in complicated and low-quality images. To tackle this problem, in the feature extraction module of the proposed FACLSTM, we assemble two decoder branches, of which one is used as normal for feature extraction and another is designed to learn additional character centre masks. These masks are expected to guide the subsequent attention module regarding where to focus. Obviously, for each time step, the attention should be focused on the centre of certain character. Moreover, these masks can also help to enhance foreground pixels and suppress background pixels.

In other works [31, 32, 65], the feature maps F and attention maps A are always combined with the element-wise multiplication $\otimes$ in the way of $F_{out} = F \otimes (1 + A)$. However, in our experiments, we find that directly concatenating feature maps F and character centre masks M can achieve better performance, since the subsequent attention-based module prefers to learn patterns from F and M directly, rather than from their fused results. Therefore, direct concatenation is used in our FACLSTM.

4.2.2 Sequential Transcription Module

As shown in Fig. 4.3, our sequential transcription module starts with an attention-equipped ConvLSTM, by which T groups of feature maps with the size of $\frac{W}{4} \times \frac{H}{4} \times C$ are generated. Here, T is the predefined maximal string length. Afterwards, a 1×1 convolutional layer is applied to reduce the feature map channels, followed by a fully connected layer and a softmax classifier that are employed to sequentially predict T characters. Details of proposed sequential transcription module are presented below.

ConvLSTM: As explained in [112], the main drawback of traditional FC-LSTM is its usage of full connections in the input-to-state and state-to-state transitions, which results in the neglect of spatial information. To retain such important information, ConvLSTM replaces all of the full connections of traditional FC-LSTM with the convolutional operations, and extends the 2-D features and states into 3-D, as shown in Fig. 4.6. Superiority of ConvLSTM over traditional FC-LSTM has been proved in [112]. Thereafter, variants of ConvLSTM have been developed for action recognition [62], object detection in video [72], gesture recognition [157, 151], etc. For example, Zhu et al. [157] combined ConvLSTM with the 3-D convolution in a multimodal model, and achieved promising gesture recognition performance. Li et al. [62] designed a motion-based attention mechanism and combined it with ConvLSTM in their VideoLSTM, which was proposed for action recognition in videos.

Key formulations of FC-LSTM can be expressed as Eq. 4.1, where $\circ$ is the Hadamard product (*i.e.*, element-wise multiplication), f denotes the activation function of input gate i_t, output gate o_t and forget gate f_t, and x_t, c_t and h_t represent input features, cell states and cell outputs, respectively.

$$\begin{aligned}
i_t &= f(w_{xi}x_t + w_{hi}h_{t-1} + w_{ci} \circ c_{t-1}),\\
f_t &= f(w_{xf}x_t + w_{hf}h_{t-1} + w_{cf} \circ c_{t-1}),\\
c_t &= f_t \circ c_{t-1} + i_t \circ tanh(w_{xc}x_t + w_{hc}h_{t-1}),\\
o_t &= f(w_{xo}x_t + w_{ho}h_{t-1} + w_{co} \circ c_t),\\
h_t &= o_t \circ tanh(c_t).
\end{aligned} \tag{4.1}$$

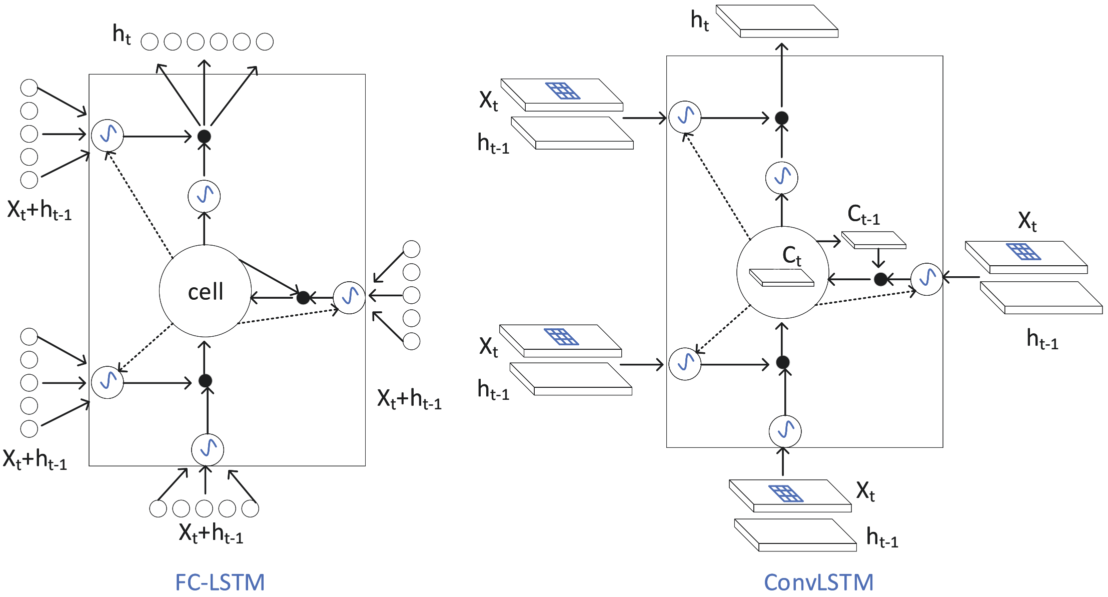


Figure 4.6 : Illustration of the FC-LSTM (left) and the ConvLSTM (right) (from [127]). The FC-LSTM is performed in 1-D space, while the ConvLSTM is performed in 2-D space.

As we can see, FC-LSTM takes 1-D sequential feature vectors as input, and calculates both the input-to-state and state-to-state transactions in a fully connected manner. Therefore, when applying it to computer vision tasks, the 2-D feature maps have to be mapped into 1-D space, during which the spatial correlation relationships among pixels are badly damaged. To take advantages of such valuable spatial and structural information in computer vision tasks, Shi et al. [112] proposed ConvLSTM by incorporating convolutional structures into LSTM. As shown in Fig. 4.3(right), all input features, gates, cell states and cell outputs are 3-D in ConvLSTM, and all of the input-to-state and state-to-state transactions are performed with the convolutional operations, instead of the fully connected ones. Thus, the key formulations of ConvLSTM can be written as Eq. 4.2, where $*$ denotes the convolutional operation.

$$
\begin{aligned}
&i_t = f(w_{xi} * x_t + w_{hi} * h_{t-1} + w_{ci} \circ c_{t-1}), \\
&f_t = f(w_{xf} * x_t + w_{hf} * h_{t-1} + w_{cf} \circ c_{t-1}), \\
&c_t = f_t \circ c_{t-1} + i_t \circ tanh(w_{xc} * x_t + w_{hc} * h_{t-1}), \\
&o_t = f(w_{xo} * x_t + w_{ho} * h_{t-1} + w_{co} \circ c_t), \\
&h_t = o_t \circ tanh(c_t).
\end{aligned}
\tag{4.2}
$$

Proposed Attention-equipped ConvLSTM: The attention mechanism has achieved excellent performance in sequential prediction tasks, such as machine translation [3], speech recognition [20], as well as scene text recognition [15, 17, 133, 109, 57]. Especially, in the field of scene text recognition, it has been widely combined with FC-LSTM or GRU to produce more accurate predictions. On the other hand, LSTM is used only for frame-level prediction in the existing works and is seldom utilized for producing sequential outputs from one single input image unless when combined with the CTC or attention mechanism.

Therefore, in this work, to adapt ConvLSTM to scene text recognition and, meanwhile, provide the proposed network location awareness, we incorporate the attention mechanism into ConvLSTM by weighting the input feature maps with attention scores derived from the cell states and cell outputs obtained at the previous time step, as illustrated in Fig. 4.7. In addition, to retain the efficiency of the proposed network, an additional bottleneck gate is assembled before the original input gate, forget gate and output gate to reduce the internal feature map channels.

Eqs. 4.3 and 4.4 provide more details on how the cell outputs and the attention scores are calculated. Here, $[\cdot, \cdot]$ is the channel-wise concatenation, $R(\cdot)$ and $S(\cdot)$ denote the ReLU activation function and the Sigmoid function, respectively, and $\widehat{x}_t$ represents the weighted inputs computed by Eq. 4.4. Keep it in mind that all of the gates $\{b, i, o, f\}_t$, inputs $\widehat{x}_t$, cell states $c_{\{t,t-1\}}$ and cell outputs $h_{\{t,t-1\}}$ in Eqs. 4.3 and 4.4 are in 3-D. Moreover, $w_{\{b,i,f,o,b2,h,x\}}$ and $bias_{\{b,i,f,o,f2,b2,y\}}$ are the involved network weights and biases, and x_t is the concatenation of feature maps F and character centre masks M produced by aforementioned encoder-decoder feature

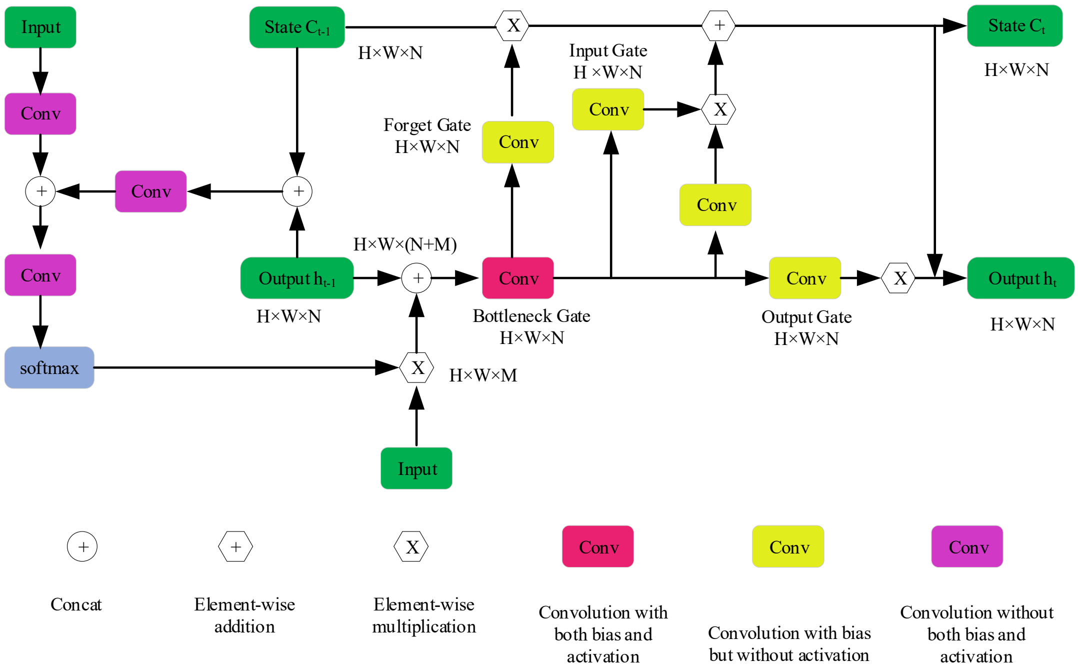


Figure 4.7 : Illustration of our proposed attention-equipped ConvLSTM (from [127]), where the inputs are weighted by attention scores derived from previous cell state and cell output.

extraction module.

$$\begin{aligned}
b_t &= R(w_b * ([\widehat{x}_t, h_{t-1}]) + bias_b),\\
i_t &= w_i * b_t + bias_i,\\
f_t &= w_f * b_t + bias_f,\\
o_t &= w_o * b_t + bias_o,\\
c_t &= S(f_t + bias_{f2}) \circ c_{t-1} + S(i_t) \circ R(w_{b2} * b_t + bias_{b2}),\\
h_t &= R(c_t) \circ S(o_t).
\end{aligned} \tag{4.3}$$

$$\begin{aligned}
h_{yt} &= [w_h * [c_{t-1}, h_{t-1}], (w_x * x)] + bias_y,\\
z_t &= w_z * tanh(h_{yt}),\\
attn_t &= softmax(z_t),\\
\widehat{x}_t &= attn_t \circ x.
\end{aligned} \tag{4.4}$$

Once the cell outputs $H = \{h_1, h_2, ..., h_T\}, h_i \in R^{M\times N\times C}$ are obtained from the proposed attention-equipped ConvLSTM, a 1×1 convolutional layer is applied to map them to $\widetilde{H} = \{\widetilde{h}_1, \widetilde{h}_2, ..., \widetilde{h}_T\}, \widetilde{h}_i \in R^{M \times N \times \widetilde{C}}$ and $\widetilde{C} < C$, which is also used to improve model's efficiency, just like the bottleneck gate does. Afterwards, a fully connected layer and a softmax classifier are designed to generate the final sequential outputs $S = \{c_1, c_2, ..., c_T\}$ from $\widetilde{H}$, where c_i is from the predefined charset. Compared with STN-OCR [7], where multiple fully connected layers and multiple softmax classifiers are assembled for sequential transcription, in our FACLSTM, only one single fully connected layer and one softmax classifier are employed and shared by T groups of feature maps.

4.2.3 Training

Loss function: The objective function L of our proposed FACLSTM consists of two parts, *i.e.*, the sequential prediction loss L_s and the mask loss L_m, as formulated in Eq. 4.5, where m, $\widetilde{m}$, $\widehat{y}$ and $\widetilde{y}$ are the ground truth masks, predicted masks, smoothed ground truth strings and predicted sequential outputs, respectively. λ is the coefficient used to balance the importance of the sequential prediction loss and the mask loss, and is set to 1 in our experiments. Additionally, the label smoothing method proposed by Szegedy et al. [119] is able to help regularize the proposed model. Therefore, given the one-hot encoded ground truth y^{OneHot}, we convert it to the smoothed version $\widehat{y}$ with Eq. 4.6. Moreover, for the ground truth masks m, we set the value of their foreground pixels (centre of characters) and background pixels to 1 and 0, respectively. Thus, the mask loss Lm is calculated in the way of Eq. 4.7.

$$L = Ls(\widehat{y}, \widetilde{y}) + \lambda Lm(m, \widetilde{m}). \tag{4.5}$$

$$\widehat{y} = (1.0 - \epsilon) * y^{OneHot} + \epsilon * (\frac{1}{N_{class}}). \tag{4.6}$$

$$Lm = 0.01 * \{1 - 2 * [\frac{\sum(m \otimes \widetilde{m})}{\sum m + \sum \widetilde{m}}]\}. \tag{4.7}$$

Generation of Ground Truth: Ground truth of character centre masks is required to optimize the proposed network. Assuming $b = (x_{min}, y_{min}, x_{max}, y_{max})$ is the bounding box of individual characters, we use the same method as that in [65] to calculate the ground truth of the corresponding mask $g = (x^g_{min}, y^g_{min}, x^g_{max}, y^g_{max})$, as shown in Eq. 4.8.

$$\begin{aligned}
&w = x_{max} - x_{min}, \\
&h = y_{max} - y_{min}, \\
&x^g_{min} = (x_min + x_max - w * r)/2, \\
&x^g_{max} = (x_min + x_max + w * r)/2, \\
&y^g_{min} = (y_min + y_max - h * r)/2, \\
&y^g_{max} = (y_min + y_max + h * r)/2.
\end{aligned} \tag{4.8}$$

Note that, the shrink ratio r is set to 0.25 in our experiments, instead of 0.5 used in [65].

4.3 Experiments

4.3.1 Datasets

We train the proposed FACLSTM network with 7 million synthetic images from SynthText dataset [38] (available at `http://www.robots.ox.ac.uk/~vgg/data/scenetext/`) without fine-tuning on individual real-word datasets, and evaluate the corresponding performance on three widely used benchmarks, including the regular text dataset IIIT5K, low-resolution and noisy text dataset SVT, and curved text dataset CUTE.

- **SynthText** is proposed by Gupta et al. [38] for scene text detection. The original dataset is composed of 800,000 scene text images, each with multiple word instances. Texts in this dataset are rendered in different styles, and annotated with character-level bounding boxes. Overall, about 7 million text images are cropped for scene text recognition.

- **IIIT5K** is built by Mishra et al. [90]. This dataset consists of 3000 text images obtained from the web. Most of these images are regular, and for individual images, two lexicons are provided, including one 50-word lexicon and one 1000-word lexicon.

- **SVT** is a very challenging dataset collected by Wang et al. [124] from the Google Street View. Totally, 647 text images with low-resolution and noise are included.

- **CUTE** is released by Risnumawan et al. [99]. There are only 288 word images in this dataset, but most of them are seriously curved. Therefore, compared with other datasets, CUTE is more challenging.

4.3.2 Implementation Details

In our experiments, all of the input images are scaled to a size of 64 × 256 with aspect ratio preserved. The maximal string length is set to 20, including one START token and one EOF token. This means up to 18 real characters are allowed within individual words. Our charset is composed of 39 characters, *i.e.*, 26 alphabet letters, 10 digits, 1 START token, 1 EOS token and 1 special token for any other symbols. The Adam optimizer with an initial learning rate of 1e-4 is employed in our work to optimize the proposed network. Totally, the proposed FACLSTM is trained for five epochs, with learning rates of 1e-4, 1e-4, 5e-5, 1e-5 and 1e-6, respectively. Moreover, the kernel size and channels (N in Fig. 4.7) of the convolutional operations in Eqs. 4.3 and 4.4 are set to 3×3 and 256, respectively. Finally, the proposed network is implemented using the Tensorflow framework.

4.3.3 Experimental Results

We evaluate the performance of our proposed FACLSTM on the aforementioned three benchmark datasets, and compare it with those of the state-of-the-art approaches. Table 4.1 presents the details of the comparison results. Note that, in this table, CA-FCN [65], ScRN [140] and SqueezedText [78] are the three latest recognisers recently published in AAAI2019, ICCV2019 and AAAI2018.

4.3.3.1 Comparison with Methods based on the Traditional FC-LSTM

As previously introduced, traditional FC-LSTM is widely used in the existing recognisers. Among methods listed in Table 4.1, RARE [108], AON [17] and

Table 4.1 : Result comparison across different methods and datasets (from [127]). Word-level recognition rate is used here. IIIT5K_No, IIIT5K_50 and IIIT5K_1k denote that no lexicon, 50-word lexicon and 1k-word lexicon are used, respectively. Smps: the number of samples used for training individual models, where * means that datasets derived from SVT are used.

Method	LSTM	Smps	IIIT5K_No	IIIT5K_50	IIIT5K_1k	SVT	CUTE
FAN [15]	FC-LSTM	12M*	87.4	99.3	97.5	**85.9**	63.9
AON [17]	FC-LSTM	12M*	87.0	**99.6**	98.1	82.8	76.8
CRNN [107]	FC-LSTM	8M*	78.2	97.6	94.4	80.8	-
(Gao et al.)* [32]	FC-LSTM	8M*	83.6	99.1	97.2	**83.9**	-
RARE [108]	FC-LSTM	8M*	81.9	96.2	93.8	81.9	59.2
R^{2}AM [57]	FC-LSTM	7M*	78.4	96.8	94.4	80.7	-
SqueezedText [78]	FC-LSTM	1M	87.0	97.0	94.1	-	-
ScRN [140]	FC-LSTM	7M	88.5	-	-	81.3	81.9
CA-FCN [65]	No	7M	**92.0**	**99.8**	**98.9**	82.1	**78.1**
(Gao et al.)* [31]	No	8M*	81.8	99.1	97.9	82.7	-
STN-OCR* [7]	No	86.0	-	-	-	79.8	-
FLSTM_base1	FC-LSTM	7M	73.7	99.0	97.4	58.7	67.4
FAFLSTM_base2	FC-LSTM	7M	87.8	99.3	98.1	78.2	75.7
FACLSTM (Our)	ConvLSTM	7M	**90.5**	99.5	**98.6**	82.2	**83.3**

FAN [15] combined FC-LSTM with the attention mechanism in the fully connected way when performing sequential transcription, while CRNN [107], R^{2}AM [57], Gao's model [32] and SqueezedText [78] utilized FC-LSTM for frame-level prediction, sequential feature encoding or other purposes. As shown in Table 4.1, our proposed FACLSTM outperforms these FC-LSTM-based methods by large margins on both regular text dataset IIIT5K (90.5% vs 87.4%) and curved text dataset CUTE (83.33% and 76.8%) when no lexicon is used. It also achieves competitive performance on IIIT5K when 1k-word lexicon and 50-word lexicon are used. Apparently, handling the text recognition task from the spatio-temporal perspective with our ConvLSTM-based FACLSTM is more effective than casting it to a sequence-to-sequence prediction problem via FC-LSTM, no matter for regular or irregular text images. Note that our FACLSTM is optimized with less training samples than most of the listed FC-LSTM-based recognisers, except for R^{2}AM [57] and SqueezedText [78], and though AON [17] is specially designed for irregular text recognition, its recognition performance on CUTE is still 6.5% lower than that of our FACLSTM.

Readers should keep in mind that apart from the 4 million training images from SynthText, the recognisers named AON [17] and FAN [15] also employed additional 8 million images provided by Jaderberg et al. [48] for their training. Jaderberg' s synthetic images are generated with a 50k-word lexicon that covers all the test words of ICDAR and SVT datasets, and blended with word images randomly-sampled from these two datasets. Thus, the recognition performance on SVT would benefit largely from the usage of Jaderberg' s images because of this strong correlation. This is also proved by Liao' s work [65], where a 4.3% accuracy improvement on SVT was achieved by their CA-FCN when additional 4 million images generated with Jaderberg' s strategy were used. In this work, to demonstrate the generalizability and robustness of proposed FACLSTM, we only employ the SynthText dataset to train our network. Therefore, to give a fair comparison, we only compare FACLSTM with recognisers not utilizing SVT-derived training images, such as CA-FCN [65] and STN-OCR [7].

4.3.3.2 Comparison with Non-LSTM based Methods

Considering the limitations of the traditional FC-LSTM on neglecting spatial and structural information and slow training convergence, CA-FCN [65], Gao's model [31] and STN-OCR [7] have also explored other non-LSTM solutions. Especially, CA-FCN [65] also addressed the recognition issue from the 2-D perspective by utilizing an FCN structure, and moreover, it used the same VGG-16 backbone and 7-million training images as our FACLSTM.

From Table 4.1, we can see that the accuracy of our proposed FACLSTM is 1.5% lower than that of the best recogniser, *i.e.,* CA-FCN [65], on the regular text dataset IIIT5K. However, on the more challenging curved text dataset CUTE, we achieve an accuracy of 83.3%, which is 5.2% higher than that of CA-FCN [65]. As for the low-resolution and noisy dataset SVT, our FACLSTM performs slightly better than CA-FCN [65] with an accuracy of 82.2% (vs. 82.1% of CA-FCN [65]). Note that, CA-FCN [65] is not an end-to-end trainable system because in order to infer the final sequential outputs from the pixel-level predictions generated by their network, an empirical rule-based word formation module is required. By contrast, our FACLSTM is able to directly produce the final sequential outputs via the proposed ConvLSTM-based sequential transcription module. Admittedly, replacing FC-LSTM with Conv-LSTM will increase GPU memory cost. Therefore, to retain the efficiency, we up-sample feature maps to a small resolution of 1/4 in the decoder branches, instead of 1/2 used in CA-FCN. Undoubtedly, this small resolution will compromise the recognition accuracy to some extent, especially for small-size and low-resolution images from the IIIT5K and SVT datasets.

4.3.3.3 Ablation Study

Furthermore, to highlight the effectiveness of our proposed focused attention module and ConvLSTM-based sequential transcription module, we compare the performance of our proposed FACLSTM with that of the following two baseline models:

- FLSTM_base1, which shares the same feature extraction module with our proposed FACLSTM, but removes the focused attention module. Besides, the sequential transcription module used in this model is the traditional attention-based FC-LSTM network, just as the one used in AON [17], FAN [15] and both Gao's models [32, 31].
- FAFLSTM_base2, which is built upon FLSTM_base1, but with the proposed focused attention module applied.

Apparently, from the comparison of FLSTM_base1, FAFLSTM_base2, we can see that the recognition accuracies on IIIT5K, SVT and CUTE datasets are elevated by 14.1%, 19.5% and 8.4%, respectively when the proposed focused attention module is assembled. As illustrated in Fig. 4.8, the focused attention module is able to accurately predict the character centre masks since it is performed in the high-level decoder branch. The significant performance improvement demonstrates that these masks are effective to help the sequential transcription module focus attention on the right character areas and suppress irrelevant background pixels. In addition, the image resolution of CUTE in much higher than that of SVT and IIIT5K and SVT is much noisier than the other two datasets. As claimed in [15, 65], the attention-based recognisers perform poorly on low-quality images because of the 'attention drift' problem, and the scene text images suffer from noisy background badly, so the accuracy improvement is more on SVT and less on CUTE when the proposed focused attention module is utilized.

Moreover, from the comparison of FAFLSTM_base2 and FACLSTM, we can see that when the traditional attention-based FC-LSTM module is replaced by our proposed attention-ConvLSTM-based sequential transcription module, further 2.7%, 3.6% and 7.6% improvements are achieved on IIIT5K, SVT and CUTE, respectively. This means that our FACLSTM is able to boost the recognition performance significantly by utilizing the proposed attention-ConvLSTM module to take benefits from the valuable spatial and structural information of text images. As clarified in [65], FC-LSTM only achieves good performance on horizontal or nearly horizontal text, and its performance on curved text is seriously limited because of the neglect of pixels' spatial correlation relationships. The huge performance improvement achieved by FACLSTM on CUTE evidences that our attention-ConvLSTM module is a good solution to this problem.

Therefore, we can say that both of the proposed focused attention module and attention-ConvLSTM module are effective. Note that the focused attention module can be removed from the network when datasets without character-level bounding box annotations are used for the training.

In summary, on the regular text dataset, our proposed FACLSTM outperforms all of listed FC-LSTM-based and non-LSTM-based recognisers, except CA-FCN, but on the more challenging curved text dataset, our FACLSTM surpasses all of the listed methods significantly with an accuracy of 83.3%, including CA-FCN (78.1%). Moreover, the comparisons with other two baseline models demonstrate the effectiveness of our proposed focused attention module and ConvLSTM-based sequential transcription module. Finally, we also give the visualization results of the predicted masks and the attention shift procedure, as shown in Fig. 4.8. The comparison results of attention predicted by FACLSTM and FLSTM base1 are shown in Fig. 4.9. Note that FACLSTM directly produces 2-D attention maps via the convolutional operations, while FLSTM base1 generates 1-D attention vectors with the fully connected layers, just as other existing FC-LSTM-based recognisers did. These 1-D attention vectors are reshaped to 2-D maps in Fig. 4.9 for an intuitional visualization. As we can see, the attention areas of FACLSTM is larger and more accurate, and the 'attention drift' problem is alleviated to some extent in our proposed FACLSTM.

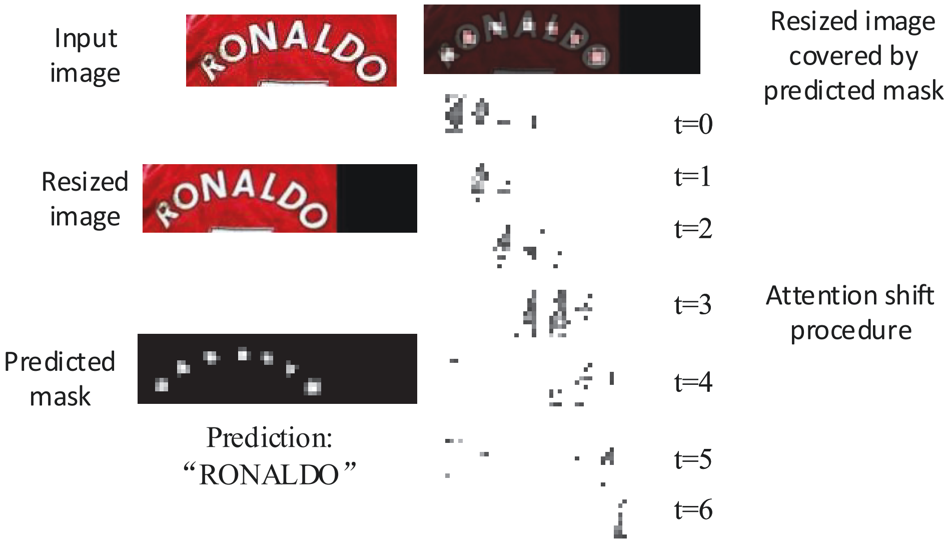


Figure 4.8 : Visualization results of predicted mask and attention shift procedure (from [127]).

4.4 Conclusion

Scene text recognition has been treated as a sequence-to-sequence prediction problem for quite a long time, and traditional FC-LSTM is widely used in current state-of-the-art recognisers. In this work, we have demonstrated that scene text recognition is actually a spatio-temporal prediction problem and we have proposed

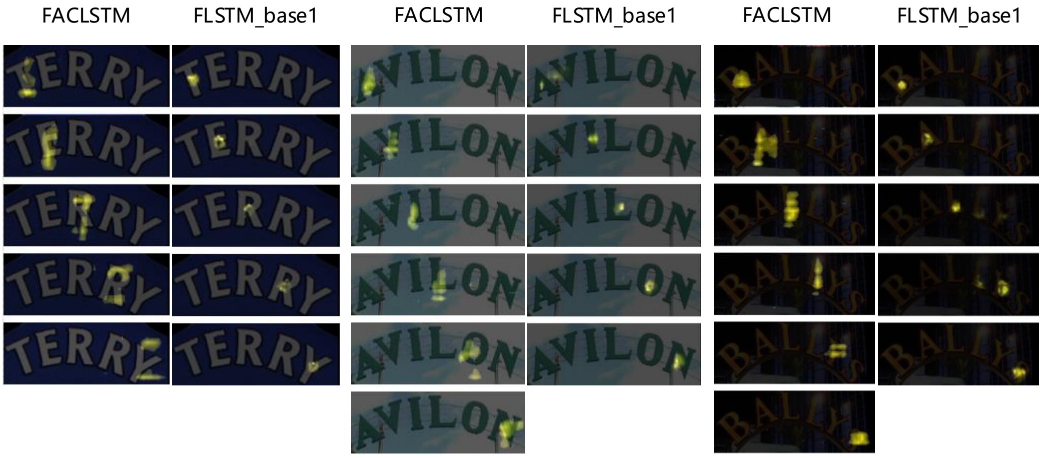


Figure 4.9 : Visualization results of attention predicted by FACLSTM and FLSTM_base1 (from [127]). Values of the attention maps are normalized and truncated for a better visualization. Note that FACLSTM directly produces 2-D attention maps, while FLSTM_base1 generates 1-D attention vectors, which are then reshaped to 2-D space.

to tackle this problem from the spatio-temporal perspective. Toward this end, we have presented an effective scene text recogniser named FACLSTM, where ConvLSTM has been applied and improved by integrating the attention mechanism in the sequential transcription module, and a focused attention module has been designed at the encoder-decoder feature extraction stage. Experimental results have revealed that, our proposed FACLSTM is able to handle both regular and irregular (low-resolution, noisy and curved) text well. Especially for the curved text, our proposed FACLSTM has outperformed other advanced approaches by large margins. Thus, we can conclude that ConvLSTM is more effective in scene text recognition than the widely used FC-LSTM since the valuable spatial and structural information can be better leveraged when performing sequential prediction with ConvLSTM.

Chapter 5

FC-LSTM-based Neural Network for Scene Text Recognition

As previously introduced, FC-LSTM has been widely used in the existing state-of-the-art text recognisers, but it cannot fully leverage the valuable spatial and structural information of 2-D images, putting an negative impact on the recognition performance. In this chapter, we propose another scene text recogniser named ReELFA, *i.e.,* Recogniser with Encoded Location and Focused Attention, to address this issue and improve the recognition performance of the existing FC-LSTM-based recognisers.

5.1 Introduction

FC-LSTM is an idea borrowed from speech recognition and machine translation, where the inputs are 1-D vectors, rather than 2-D feature maps. Therefore, to adapt FC-LSTM to scene text recognition, the most straightforward way is pooling or flattening 2-D feature maps into 1-D space, as shown in Fig. 5.1. However, the pooling operation will result in a loss of vertical information and the flattening operation will disturb spatial relationships of pixels. Therefore, as claimed in [65], the existing FC-LSTM-based models can only achieve good performance on horizontal or nearly horizontal texts. As for curved or skewed text, the performance is far from satisfactory.

Though ConvLSTM can better utilize the spatial information of images in computer vision tasks, it requires more GPU memory. Therefore, in our last recogniser, *i.e.,* FACLSTM, feature maps are down-sampled to a smaller resolution to retain models' efficiency. This strategy works well for text images with high resolution and achieves promising performance on shape curved text images, but, for low-

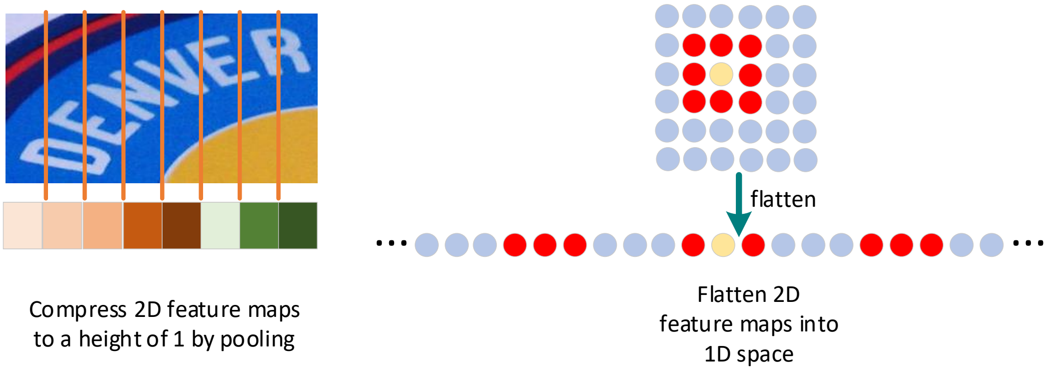


Figure 5.1 : Converting 2D feature maps into 1D space to adapt FC-LSTM to scene text recognition (from [125]).

resolution images, the down-sampling operation has affected the recognition performance to some extent.

On the other hand, most of the existing state-of-the-art text recognisers are on the base of FC-LSTM. Therefore, an efficient strategy to exploit the spatial information in FC-LSTM-based recognisers is meaningful. Toward this end, in this work, we propose to utilize the one-hot encoded location to indicate the spatial relationships of individual pixels. This idea is motivated by Wojna's work [133], where the location information of pixels is utilized to tackle the permutation invariant problem of the widely used spatial attention mechanism. The proposed idea is efficient and effective, and can be easily incorporated into any other existing FC-LSTM-based recognisers. Details of our newly designed scene text recogniser, *i.e.*, ReELFA, are presented below.

5.2 Proposed Method

As illustrated in Fig. 5.2, our proposed ReELFA consists of two modules, *i.e.*, an encoder-decoder feature extraction module, which is the same as the one used in FACLSTM (see Sec. 4.2.1 for more details), and an attention-LSTM-based sequence transcription module. The differences between ReELFA and aforementioned FA-

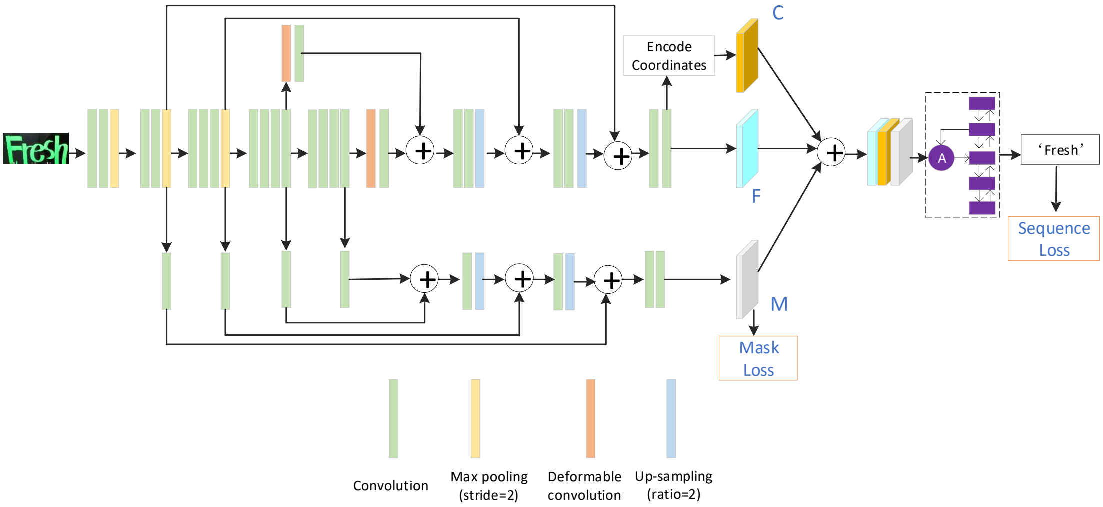


Figure 5.2 : The structure of our proposed ReELFA network (from [125]).

CLSTM are three-folds. Firstly, feature maps extracted by the VGG-16 backbone are up-sampled to a larger resolution of $\frac{1}{2}$ in ReELFA, instead of $\frac{1}{4}$ used in FA-CLSTM. Secondly, extra one-hot encoded locations are attached after the feature maps extracted by VGG-16 and the character centre masks generated by the focused attention module to take advantage of pixels' spatial relationships. Finally, the attention-equipped FC-LSTM, rather than ConvLSTM, is used for the sequence transcription.

5.2.1 One-hot Encoded Location

As mentioned previously, pooling and flattening are the two most popular ways to project 2-D feature maps into 1-D space. The pooling operation results in unrecoverable loss of spatial information, while the flattening operation just disrupts orders of pixels. Therefore, in this work, the flattening operation is adopted.

To indicate the spatial relationships of pixels in feature maps, inspired by Wojna's work [133], we propose to utilize the one-hot encoded coordinates to make the FC-LSTM 'location aware'. As shown in Fig. 5.3, we attach the predicted character centre masks m_k and one-hot encoded coordinates c_k after the extracted feature maps f_k in the channel dimension. In addition, given pixels P_1 and P_4, and P_1's

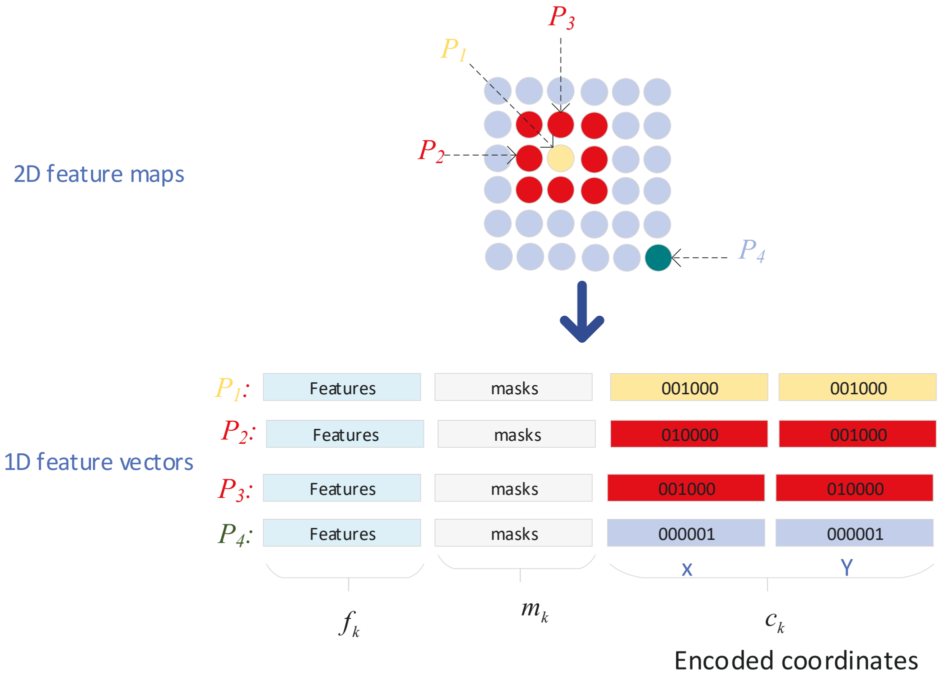

P3
P1
P2
2D feature maps
P4
P1: Features masks 001000 001000
P2: Features masks 010000 001000
1D feature vectors
P3: Features masks 001000 010000
P4: Features masks 000001 000001
X
Y
f_k
m_k
c_k
Encoded coordinates

Figure 5.3 : Illustration of the proposed one-hot encoded location (from [125]).

adjacent pixels P_2 and P_3, we can see that the encoded coordinates c_k of pixel P_1 is much closer to that of its adjacent pixels P_2 and P_3 when comparing with P_4, which has longer distance to P_1 than other pixels.

5.2.2 Attention-LSTM-based Sequence Transcription

At the end of proposed ReELFA, an attention-LSTM-based sequence transcription model is exploited to generate target sequential outputs $(y_1, y_2, ..., y_N)$ from the input feature vectors $[f_1, f_2, ..., f_K]$, the centre masks $[m_1, m_2, ..., m_K]$ and the encoded coordinates $[c_1, c_2, ..., c_K]$, where $f_k \in R^L$, $m_k \in R^H$ and $c_k \in R^T$, and K is the length of sequential feature vectors. The procedure can be formulated in Eq. 5.1, where u_t and $\widetilde{y}_t$ are the weighted features and the expected prediction results at time t, and x_t, o_t and s_t denote the inputs, outputs and states of the FC-LSTM at time t, respectively. Moreover, y_{t-1} represents the ground truth y_{t-1} at the training stage, and equals to the prediction result $\widetilde{y}_{t-1}$ at the inference stage. Additionally, the attentions of the k^{th} feature vector at time t are denoted by $\alpha_{t,k}$, and can be derived from Eq. 5.2. Here, V_a is a vector and outputs of $tanh$ is applied element-wise to V_a.

$$\begin{aligned} u_t &= \sum_{k=1}^{K} \alpha_{t,k}(f_k + c_k + m_k), \\ x_t &= W_y \overline{y}_{t-1} + W_{u1} u_{t-1}, \\ (o_t, s_t) &= LSTM(x_t, s_{t-1}), \\ \widetilde{o_t} &= softmax(W_o o_t + W_{u2} u_t), \\ \widetilde{y_t} &= \arg\max_y \widetilde{o_t}(y). \end{aligned} \tag{5.1}$$

$$\begin{aligned} a_{t,k} &= V_a^T tanh(W_s s_t + W_f f_k + W_c c_k + W_m m_k), \\ \alpha_t &= softmax_k(a_{t,k}). \end{aligned} \tag{5.2}$$

5.3 Experiments

5.3.1 Comparison with Other Scene Text Recognisers

The same loss configuration and training protocol as FACLSTM are used in this work to optimize the proposed ReELFA. Note that apart from aforementioned IIIT5K, SVT and CUTE, an additional dataset named IC15, short for ICDAR 2015, is used in this section to evaluate the performance of ReELFA on low-resolution images. IC15 proposed in [52] consists of about 2,077 scene text images, including 200 irregular ones (arbitrary-oriented, curved or perspective). Comparison results are presented in Table 5.1.

Comparison with Attention-LSTM-based Models: Among the methods listed in Table 5.1, RARE [108], AON [17] and FAN [15] use the combination of bi-LSTM and attention mechanism in the sequence transcription module. FAN and AON are more recent works than RARE, while RARE and AON are specially designed for irregular text recognition. Additionally, a focusing network is designed in FAN to tackle the problem of 'attention drift'. Moreover, our ReELFA and RARE [108] are trained with only SynthText dataset, while FAN and AON are trained with both SynthText dataset and Jaderberg's dataset [48].

Table 5.1 : Results obtained by different methods. 'IIIT5K_*' indicates the lexicon type used for the evaluation of the IIIT5K dataset. 'Ours_noEL' and 'Ours_noFA' represent our model without the encoded location and focused attention respectively. '*' means that the word images containing non-alphanumeric characters are removed from the test dataset. *_bi means binary network setting.

Methods	**IIIT5K_No**	**IIIT5K_50**	**IIIT5K_1k**	**SVT**	**CUTE**	**IC15**
FAN [15]	87.4	99.3	97.5	**85.9**	63.9	66.2
AON [17]	87.0	*99.6*	*98.1*	82.8	76.8	*68.2*
CRNN [107]	78.2	97.6	94.4	80.8	-	-
(Gao et al.)* [32]	83.6	99.1	97.2	*83.9*	-	-
(Gao et al.)* [31]	81.8	99.1	97.9	82.7	-	-
RARE [108]	81.9	96.2	93.8	81.9	59.2	-
STN-OCR* [7]	86.0	-	-	79.8	-	-
SqueezedText_bi [78]	86.6	96.9	94.3	-	-	-
SqueezedText(full-precision) [78]	87.0	97.0	94.1	-	-	-
R^{2}AM [57]	78.4	96.8	94.4	80.7	-	-
CA-FCN [65]	**92.0**	**99.8**	**98.9**	82.1	78.1	-
ScRN [140]	88.5	-	-	81.3	81.9	-
FACLSTM [127]	90.5	99.5	98.6	82.2	83.3	-
Ours_noEL	87.8	99.3	*98.1*	78.2	75.7	66.6
Ours_noFA	89.8	99.2	97.9	79.8	*81.6*	66.9
ReELFA (proposed)	*90.9*	99.2	*98.1*	82.7	**82.3**	**68.5**

From Table 5.1, we can see that RARE [108] is significantly surpassed by FAN [15], AON [17] and our proposed ReELFA on all datasets, and FAN [15] achieves the best performance on the SVT dataset. However, on the regular and curved text datasets IIIT5K and CUTE, our proposed ReELFA achieves the best performance, even without assistance from Jaderberg's dataset [48]. Especially, on the CUTE dataset, we get an accuracy of 82.3%, which is 17.4% and 5.5% higher than FAN and AON, respectively. Therefore, our proposed ReELFA is more robust to curved text recognition. As for the IC15 dataset, our proposed model obtains the best performance of 68.5%, which is slightly better than AON's [17] 68.2%. It is notable that FAN and AON take advantages of the prior knowledge of ICDAR and SVT datasets by leveraging Jaderberg's 4 million samples [48].

Comparison with Other Models: Methods without using attention-LSTM also achieve promising performance in the field of scene text recognition, as shown in Table 5.1. In these methods, R^{2}AM [57], CRNN [107], CA-FCN [65] and both Gao's methods [32, 31] are trained with SynthText dataset, while SqueezedText [78] and STN-OCR [7] are trained with text images generated by new rending engines.

Apparently, CA-FCN [65] and our proposed ReELFA are on the first and second places on the IIIT5K dataset with accuracies of 92.0% and 90.9%, respectively, which outperform other methods significantly. For the low-resolution and noisy dataset SVT, even though Gao et al. [32] reported a higher accuracy of 83.9%, we cannot say their model is more robust than CA-FCN [65] and ours because their model is evaluated on an incomplete dataset, where word images containing non-alphanumeric characters or with less than three characters are removed. Finally, on the challenging curved text dataset CUTE, our ReELFA achieves the best performance of 82.3%, which is 4.2% higher than CA-FCN [65].

Compared with our previously proposed ConvLSTM-based scene text recognizer, i.e., FACLSTM, we can find that our FC-LSTM-based recognizer and ConvLSTM-based recognizer can achieve similar performances on regular texts and noisy texts, but on curved texts, the ConvLSTM-based recognizer outperforms the FC-LSTM-based recognizer obviously, which means ConvLSTM can leverage spatial and structural information better than FC-LSTM. However, according to our experiments, ReELFA can achieve faster convergence speed and inference speed than FACLSTM, which demonstrates the efficiency of FC-LSTM.

The Importance of EL&FA: To highlight the importance of proposed encoded location and focused attention modules, we also conduct ablation experiments on two baseline models. The first one named 'Ours_noEL' in Table 5.1 is the version without encoded location module and the second one named 'Ours_noFA' is the version without focused attention module. The rest of these two baseline models' configurations are just the same as our ReELFA.

From Table 5.1, we can see that when the one-hot encoded location module is dropped, the accuracies on IIIT5K, SVT, CUTE and IC15 datasets are decreased by 3.1% (from 90.9% to 87.8%), 4.5% (from 82.7% to 78.2%), 6.6% (from 82.3% to

75.7%) and 1.9% (from 68.5% to 66.6%), respectively. The significant performance degradation evidences the importance of spatial correlation information to scene text recognition, especially to curved text recognition, and the effectiveness of proposed strategy.

Additionally, when the attention focusing module is removed, the performance on IIIT5K, SVT, CUTE and IC15 datasets dropped by 1.1% (from 90.9% to 89.8%), 2.9% (from 82.7% to 79.8%), 0.7% (from 82.3% to 81.6%) and 1.6% (from 68.5% to 66.9), respectively. Though the performance gap between 'Ours_noFA' and our proposed ReELFA is not as large as that between 'Ours_noEL' and ReELFA, the recognition accuracies on both regular and irregular texts are improved in certain degrees when the focused attention module is deployed. Therefore, the current attention-based models do suffer from the 'attention drift' problem, which can be alleviated by focusing attentions on the centres of characters.

5.3.2 End-to-End Scene Text Reading System

We combine ReELFA with aforementioned scene text detector named DeepText to form an end-to-end trainable scene text reading system, and apply this system to licence plate recognition. Fig. 5.4 shows the framework of the proposed system. As we can see, the detected text areas are cropped as the inputs of the followed recogniser directly, and the detector and the recogniser are trained simultaneously with the same optimizer and learning rate.

The proposed system is trained with 1,000 labelled images collected from Westfield car park and tested on 3,000 images. Its recognition performance is 93.4%, which is comparable with the recognition performance (93.2%) of the commercial OCR software of Westfield company.

5.4 Conclusion

In this work, we have proposed another scene text recogniser named ReELFA to take advantage of the spatial correlation information of pixels via the one-hot encoded location. The proposed strategy is efficient and effective, and can be in-

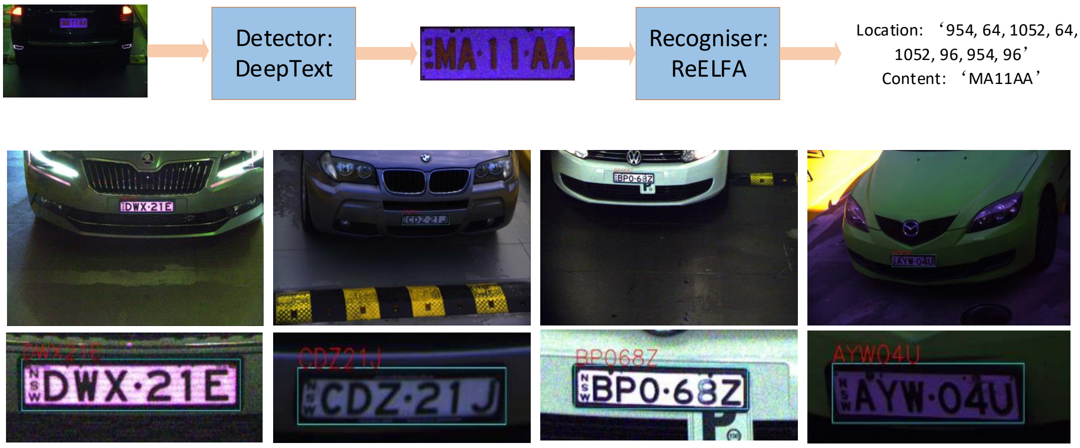


Figure 5.4 : End-to-end trainable scene text reading system and related results on licence plate recognition.

tegrated into any other existing FC-LSTM-based recognisers easily. Experimental results conducted on IIIT5K, SVT, IC15 and CUTE datasets have demonstrated that the proposed ReELFA is able to achieve comparable performance on the regular, low-resolution and noisy text datasets, and outperforms the existing state-of-the-art approaches on the more challenging curved text dataset. In addition, the ablation study has shown that, with the assistance of proposed focused attention module and one-hot encoded location module, performance of the popular FC-LSTM-based recogniser is significantly improved.

Chapter 6

Flexible SPP for CNN-based LSTM-free Handwritten Text Recognition

The same as scene text recognition, handwritten text recognition has also attracted intensive attentions in recent years because of its vast applications in both industrial projects and financial transactions, such as mail sorting system, bank cheque processing, etc. Unlike scene texts, handwritten texts are usually with uniform appearances and captured under controlled conditions, and their backgrounds are not as complicated as those of scene texts. However, because of the intractable connected characters and various writing styles, handwritten text recognition is still a challenging task. Especially, handwritten text recognition has been struggling with connected patterns fiercely in last decades.

In literature, over-segmentation and segmentation-free frameworks are commonly applied to handle the handwritten text recognition task. For the past years, RNN/LSTM combining with CTC has occupied the domain of segmentation-free handwritten text recognition, while CNN is just employed as a single character recogniser in the over-segmentation framework. The main challenges for CNN to directly recognize handwritten texts are the appropriate processing of arbitrary input string length, which implies arbitrary input image size, and reasonable design of the output layer. In this work, we propose a sequence labelling convolutional network [128], *i.e.,* CFSPP, for the recognition of handwritten texts, especially, the connected patterns. We properly design the structure of the network to predict how many characters present in the input images and what exactly they are at every position. Moreover, Spatial pyramid pooling (SPP) is utilized in our network with a new implementation to handle arbitrary string length, and we also propose a more flexible pooling strategy called FSPP to adapt the network to the straightforward recognition of long strings better. To demonstrate the superiority of our proposed CFSPP, we conduct experiments on two benchmark handwritten digital string datasets and our own cell-phone number dataset named PhPAIS.

6.1 Introduction

Because of our writing habits, strokes of adjacent written characters are usually connected to each other, introducing huge challenges to the task of handwritten text recognition. A straightforward solution is to segment text images into components corresponding to single characters or part of characters, followed by analysing the recognition results of each component or their combination to infer the optimal integrated results. In literature, this pipeline is called over-segmentation framework. For methods developed under this framework, classifiers are trained only for single character classification. Therefore, at the initial stages, it is expected to form intact single characters by over segmenting strokes and combing the consecutive ones. At the beginning of deep learning era, to take advantage of deep networks, CNN is usually employed as single character classifiers. Though the recognition accuracy has been improved significantly, performance of the entire system is still far from satisfactory because of limitations of other components.

The alternative solution resorts to segmentation-free framework, which directly produces sequential character predictions from the entire input text images via end-to-end trainable networks without segmentation. For example, some researchers try to obviate segmenting by utilizing RNN/LSTM combining with CTC in the task of handwritten text recognition. This strategy has been flourishing since the revival of deep learning and has achieved promising performance in the ICFHR2014 HDSR competition [26]. In addition, handwritten text recognisers with segmentation-free framework have been widely applied in various languages such as English [36], Arabic [36], Chinese [88], etc. It is remarkable that, on the ICFHR2016 handwritten text recognition competition [101], all of the six submissions are RNN/LSTM/CTC-based. In these recognisers, RNN maps inputs of current time step to corresponding outputs with considering its historical information, *i.e.,* outputs of previous time step, and CTC is assembled at the end of the networks to perform sequence labelling without requirement of pre-segmenting inputs and post-processing network’ s outputs. Moreover, to cope with the gradient vanishing problem, LSTM is proposed as an improvement of RNN and has replaced RNN in all of the subsequent recognisers. In summary, according to literature, over-segmentation recognisers usually require sophisticated techniques on over segmenting, touch splitting, single character recognizing and best path searching, while segmentation-free recognisers are with more convenient end-to-end trainable structures. Currently, recognisers with segmentation-free framework have surpassed the traditional over-segmentation-based ones and become the dominated solutions to handwritten text recognition.

CNN is a popular network structure in the field of computer vision and has played critical roles in many tasks. Numerous sophisticated techniques are studied to promote its expression ability and recognition accuracy. However, few strategies are proposed for sequence labelling with CNN except the one proposed by Goodfellow et al. [34]. The main difficulties are posed by the arbitrary input size and the reasonable structure design. Goodfellow et al. [34] recognized Street View House Number (SVHN) with CNN by assembling multiple softmax classifiers at the output layer. The first one was employed to predict string length and its prediction result determined how many following classifiers should be taken to figure out target string. But this method separately trained individual classifiers, and saved a separate weight matrix for each separate digit classifier. As discussed in [34], for long sequences this could incur too high of a memory cost. Moreover, this method resized input images into a fixed size and alleviated deformation caused by resizing with additional background pixels. This operation also limited the capability of this method. As pointed out by Goodfellow et al. [34], for large maximum length, their method was unlikely to scale well. Besides, the resizing operation is not suitable for handwritten string recognition since the lengths of handwritten strings are significantly different, and any resizing operation can cause serious deformation of characters. In this case, easing deformation by padding additional background pixels will result in heavy computation burden. SPP proposed by He et al. [41] has the capability to handle the arbitrary input sizes without resizing operation. Unfortunately, in order to take advantage of the existing GPU implementations (such as *cuda-covnet* and *caffe*), He et al. [41] implemented their SPP network by two fixed-size networks sharing parameters and preserved the SPP behaviours by multiple conventional sliding window pooling layers with different pooling sizes and strides. Hence, during the training procedure, training samples still needed to be resized into fixed sizes.

In this work, we carefully design a sequence labelling convolutional network, *i.e.,* CFSPP, to recognize handwritten texts, specially, the intractable connected patterns. SPP is employed in our network with a new implementation so that training samples even in the same batch can own different sizes. For the original SPP, images with different aspect ratios, which indicate different string lengths, share the same scale setting. Intuitively, more features should be extracted for images containing more characters, while fewer features are enough for those with fewer characters. Towards this end, we propose a more flexible pooling strategy called FSPP, which fixes the vertical scale setting and designs different horizon scale setting to images according to their aspect ratios.

To demonstrate the effectiveness of the proposed network, we conduct experiments on handwritten digit strings from CVL and ORAND-CAR datasets. Comparison with all the participating methods of ICFHR2014 competition shows the superiority of our proposed network. Furthermore, we collect a cell-phone number dataset named PhPAIS (cell-phone number dataset collected by Lab of Pattern Analysis and Intelligence System) from the real China post images to further verify the practicability of our proposed CFSPP.

6.2 Proposed Method

The architecture of our proposed recogniser is depicted in Fig. 6.1. As we can see, it is a CNN-based LSTM-free network that is composed of three components, *i.e.,* CNN-based feature extractor, SPP-based feature converter and model

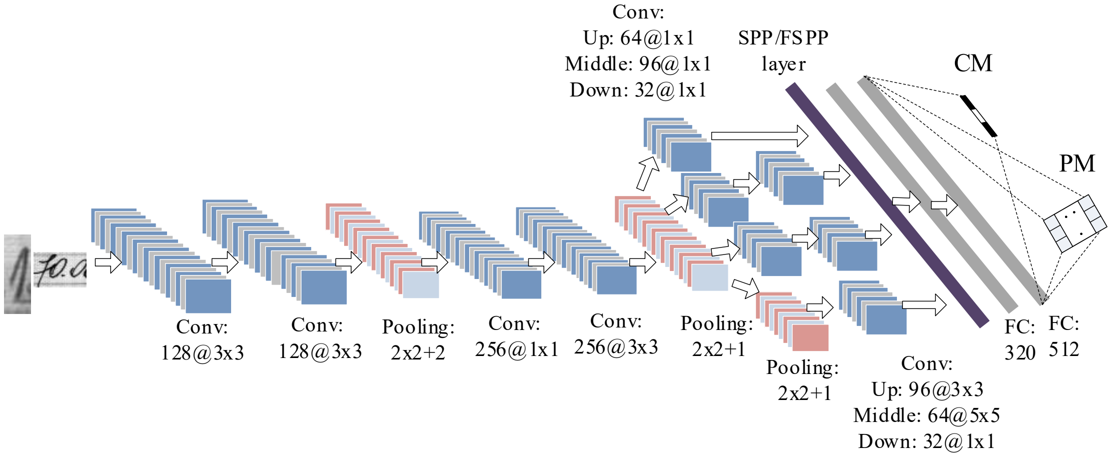


Figure 6.1 : Structure of proposed method (from [128]).

predictor. In particular, we firstly design four convolutional layers and two mean pooling layers before an Inception module to extract expressive features from input text images. Then, an SPP layer or a Flexible SPP (FSPP) layer is embedded to convert feature maps with arbitrary sizes to feature vector with fixed lengths so that they can be fed into subsequent fully-connected layers. If we regard SPP or FSPP layer as extracting global features at multiple scales, and Inception module as extracting local features at multiple scales, setting a SPP or FSPP layer after an Inception module [118] will be beneficial for the model to extract richer information. Afterwards, outputs of the fully connected layer are fed into the final task-specific module to produce the expected sequential characters. Intuitively, information from two aspects needs to be extracted: (1) how many characters are contained in the input images, and (2) what exactly it is at each position. Therefore, we equip the proposed network with two modules named CM, which is short for ‘counting module’, and PM, which is short for ‘prediction module’, to predict the corresponding information. Apparently, the proposed network is supposed to have a cluster of classifiers rather than a single one. Therefore, we need to blend prediction errors of these classifiers into one structure when we design the cost function for the network.

6.2.1 CNN-based Feature Extraction

The existing recognisers usually rescale input text images to a fixed size to fit the requirement of computation and network structures. Moreover, to avoid deformation caused by resizing, additional background pixels are usually padded around text images. As shown in Fig. 6.2, when the padding operation is not applied, huge deformation will be introduced because of the large differences in string lengths, while when additional pixels are padded, the computation burden will increase significantly since all of the pixels are treated equally. To tackle this problem, in our work, we firstly pad images to the maximal image size of current batch, and then flow the original heights and widths of individual text images to GPUs together with the padded images. According to the original heights and widths of input images, GPUs do not perform any calculation for padded background pixels so that extra computation can be saved.

In Fig. 6.1, $a@b \times c$ means that there are a filters with size $b \times c$ in current convolution layer and $d \times e + f$ means that current average pooling layer is with a pooling window size of $d \times e$ and a step of f. As seen, in our recogniser, text images are firstly sent to four convolutional layers and two pooling layers, and then an Inception module is employed to extract features from multiple scales. The Inception module is a carefully hand-crafted structure proposed in GoogleNet [118] and consists of multiple parallel branches with various receptive fields. As clarified in [118], this module is used to approximate an optimal local sparse structure with

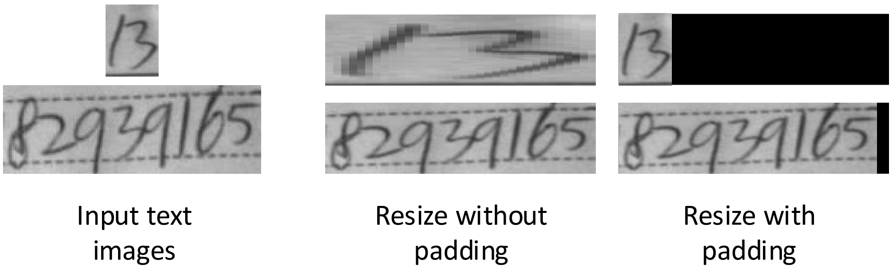


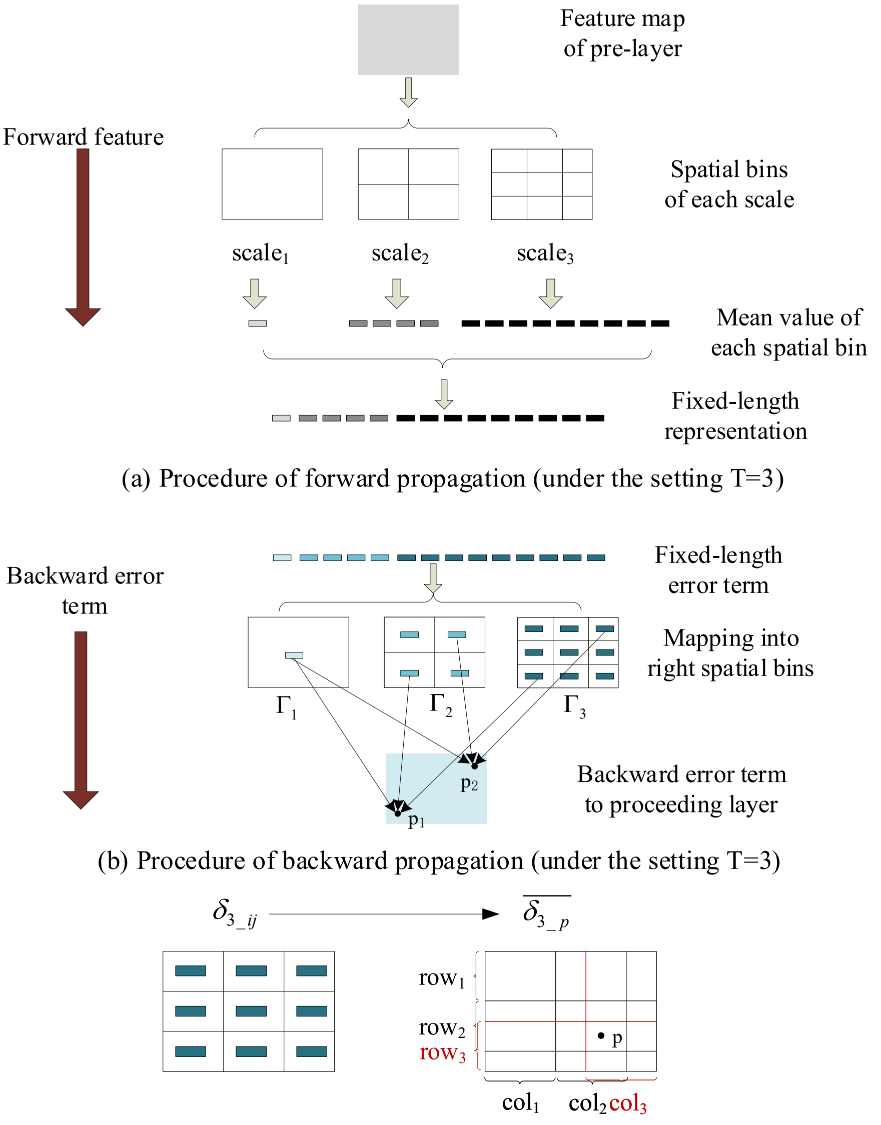
Figure 6.2 : Resize text images with/without padding.

readily available dense components.

6.2.2 Reimplementation of SPP

SPP is proposed by He et al. [41] to handle objects with arbitrary sizes in the object detection task. Expectedly, it can be applied on images with any sizes. However, to alleviate the existing GPU implementations (e.g., *cuda-convnet* and *caffe*) that run on images with fixed sizes, in [41], SPP is implemented with two fixed-size networks that share parameters among layers with different configurations. Furthermore, to adapt to this implementation, He et al. [41] specially designed two training strategies in their work, *i.e.,* single-size training and multi-size training, where input images still needed to be rescaled into fixed sizes. Apparently, as explained above, this implementation is not suitable for handwritten text recognition. Therefore, in this work, we reimplement SPP in a more straightforward way as presented below.

The input of the SPP layer is a serial of feature maps, each of which needs to be pooled into fixed-length representations at this layer in the way depicted in Fig. 6.3(a). Assuming the total number of pre-defined scales is T, an input feature map is divided into different bins evenly at each scale, and for the t^{th} scale, there will be $t \times t = t^2$ bins. Then, for each bin, we calculate the mean value or max value (mean value is chosen in our experiments) as its representation. Obviously, if we set the overall scale number to be T, $1 + 2^2 + ... + T^2 = \frac{T(T+1)(2T+1)}{6}$ features will be obtained for each feature map. We denote the size of an input feature map as $a \times b$. For the t^{th} scale, if a or b cannot be exactly divided by t, bins of the last column or row will share pixels with the previous ones.



(a) Procedure of forward propagation (under the setting T=3)

(b) Procedure of backward propagation (under the setting T=3)

(c) Error term contribution of $\Gamma_3$to point p

Figure 6.3 : Implementation of spatial pyramid pooling (from [128]).

The procedure of back propagating error terms is described in Figure 6.3(b). Firstly, elements of error term vector are mapped into corresponding spatial bins of each scale. Then, error terms of proceeding layer are calculated from values of bins at different scales. For the t^{th} scale, we denote error term of bin at position (i, j) as δ_{t_ij}, and the error term map of this scale as Γ_t. $\overline{\delta_{t\_p}}$ means the total error term contribution of bins at t^{th} scale to pixel p of certain feature map in previous layer. The final error term of pixel p is calculated by Eq. 6.1.

$$\delta p = \sum_{i=1}^{T} \frac{\overline{\delta_{i\_p}}}{\lfloor a/i \rfloor \times \lfloor b/i \rfloor}. \tag{6.1}$$

Note that in the forward procedure, several bins may share the same pixels. Therefore, error terms are also shared by multiple pixels in the corresponding backward procedure. Figure 6.3(c) demonstrates one of the sharing cases. In this situation, $\overline{\delta_{3\_p}}$ is computed by Eq. 6.2 since there are four bins sharing pixel p in the forward procedure.

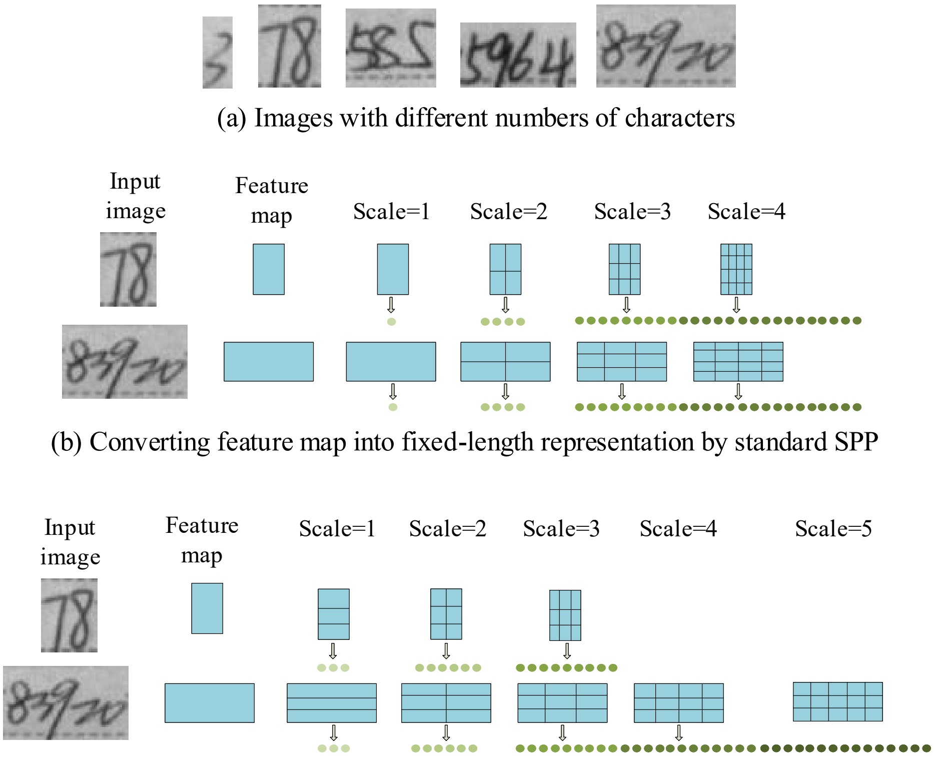
$$\overline{\delta_{3\_p}} = \delta_{3\_22} + \delta_{3\_33} + \delta_{3\_32} + \delta_{3\_23}. \tag{6.2}$$

With above implementation, our network can be trained with samples in arbitrary sizes directly. In the experiments, samples within the same batch may have different sizes, unlike the implementation in [41]. Theoretically, our implementation can be used to recognize handwritten strings with any length, while the original training strategy proposed in [41] cannot since its resizing operation will cause deformation of strings. The longer the string is, the severer the deformation becomes.

6.2.3 Flexible SPP Layer

Intuitively, text images containing longer strings should be represented by more features, while those with shorter strings only need fewer features. However, SPP or any other existing networks treat all of the input images equally and extract features with the same sizes for them. To address this issue, in this work, we modify SPP to a more flexible version, *i.e.,* Flexible SPP (FSPP), to extract features with different sizes for text images containing different strings.

Flexible SPP (FSPP) is different from SPP in the following two aspects: (1) in the vertical direction, the scale is fixed (set to 3 in our work), and (2) in the horizontal direction, the setting of scale is dynamically changed according to the aspect ratios of input images. The procedure of feature extraction is shown in Fig. 6.4. As seen, in SPP, 30 features are extracted for both text images containing '78' and '83920' when the sale is set to 4. While in the FSPP strategy, 18 features are extracted for the former and 45 features are extracted for the latter when the scale is set to 3 and 5 for them, respectively.



(a) Images with different numbers of characters

(b) Converting feature map into fixed-length representation by standard SPP

(c) Converting feature map into fixed-length representation by the proposed FSPP

Figure 6.4 : Feature extraction of the SPP layer and FSPP layer from feature maps with arbitrary sizes (from [128]).

Then, we send the extracted features into the subsequent fully-connected layer in the way shown in Fig. 6.5, where FM_i represents the converted features of i^{th} feature map. In this figure, the scale is set to 2, 3 and 4 (or larger) for images with aspect ratios (denoted by AR) within the interval $(0, \tau_1]$, $(\tau_1, \tau_2]$ and (τ_2, ∞), respectively. Each node Nd in the next layer needs to connect with all of the extracted features in the FSPP layer. For example, if the aspect ratio of certain input image is within $(\tau_1, \tau_2]$, the scale will be set to 3 in the FSPP layer, and each node Nd will connect with $3 \times (1 + 2 + 3) \times M = 18M$ features totally. In practice, we need the image sizes together with raw image pixels flow into the network simultaneously so that GPUs can directly determine pooling scales for

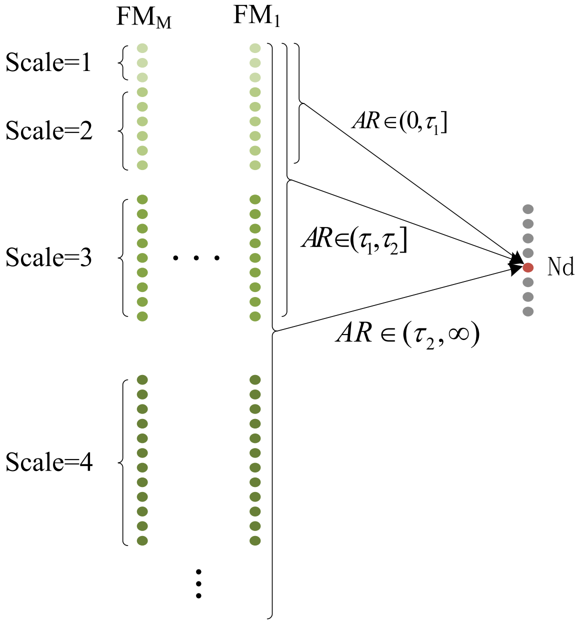


Figure 6.5 : Connection of the FSPP layer to the next fully-connected layer (from [128]).

individual images at this layer.

6.2.4 Model Prediction

Considering an input image X containing N characters $D = \{d_1, d_2, ..., d_N\}$, the objective of the basic approach is to predict N and digit d_i at position p_i correctly. Suppose the maximal input sequence length is K, which is pre-determined, we design K nodes at CM and K softmax classifiers at PM. Let us denote the prediction result of CM as L, $L \leqslant K$, and K prediction results of PM as $S = \{s_1, ..., s_K\}, s_i \in \Phi \cup \{NAN\}$, where Φ represents the alphabet of characters needed to be predicted and NAN means that there is no character showing at this position. In this architecture, we assume that characters are independent from each other. Therefore, for a specific sequence with l characters $G = \{l, g_1, ..., g_l\}, l \leq K$, its probability output can be defined as Eq. 6.3, where $p(\cdot)$ is the probability output of softmax classifiers. The optimal prediction result $G' = \{l', g'_1, ..., g'_{l'}\}$ of the architecture is determined by Eq. 6.4.

$$P(G|X) = p(L = l|X) \prod_{i=1}^{l} p(s_i = g_i|X) \prod_{j=l+1}^{K} p(s_j = NAN|X). \tag{6.3}$$

$$G' = (l', g'_1, ..., g'_{l'}) = \underset{l, g_1, ..., g_l}{\operatorname{argmax}} \, logP(G|X). \tag{6.4}$$

At the backward propagating stage, the gradients are back propagated with Stochastic Gradient Descent (SGD) with momentum. We denote the batch size as m, then the cost function of the proposed network can be formulated as Eq. 6.5, where $X^{(i)}$ is the input image of the i^{th} sample, which contains $N^{(i)}$ characters $\{d_1^{(i)}, ..., d_{N^{(i)}}^{(i)}\}$, and $s_k^{(i)}$ denotes the prediction result of the k^{th} classifier in PM for the i^{th} sample. This cost function fuses prediction errors of CM and PM into one structure, so we can train all of the softmax classifiers simultaneously by minimizing this function. In order to prevent overfitting, we introduce a weight decay term and rewrite the cost function to Eq. 6.6, where λ is the weight decay used to control the relative importance of this item, and W denotes the set of weight coefficients of the network. Parameters θ including weight coefficients W and bias b are updated by $\theta = \theta - \alpha \nabla_\theta J$, where α is the learning rate.

$$\begin{split} J = -\frac{1}{m}[\sum_{i=1}^{m} log(p(L = N^{(i)}|X^{(i)}) \prod_{k=1}^{N^{(i)}} p(s_k^{(i)} = d_k^{(i)}|X^{(i)}) \\ \prod_{k=N^{(i)}+1}^{K} p(s_k^{(i)} = NAN|X^{(i)}))]. \end{split} \tag{6.5}$$

$$\begin{split} J = -\frac{1}{m}[\sum_{i=1}^{m} log(p(L = N^{(i)}|X^{(i)}) \prod_{k=1}^{N^{(i)}} p(s_k^{(i)} = d_k^{(i)}|X^{(i)}) \\ \prod_{k=N^{(i)}+1}^{K} p(s_k^{(i)} = NAN|X^{(i)}))] + \frac{\lambda}{2} \sum w^2, \\ s.t. \quad w \in W. \end{split} \tag{6.6}$$

In our design, the possibilities of strings with length from 1 to K are calculated and compared while considering all of the classifiers' prediction results. While in

Goodfellow's design [34], if the prediction result of the first classifier is k, the prediction results of the subsequent k classifiers are simply concatenated to obtain the target string, and the prediction results of other classifiers are omitted. Moreover, as discussed in [34], given a maximal string length setting K, Goodfellow's model [34] employs K classifiers, each of which has its own weight matrix, to predict characters at individual locations, resulting in a dramatic increase of memory cost. Therefore, their model is not applicable for text images with long strings. By contrast, our model predicts characters at all locations with only one single character classifier, so it does not suffer from the memory cost problem for any maximal string length settings. In addition, unlike Goodfellow's work [34], which resizes input images into a fixed size, we utilize the re-implemented SPP or FSPP to handle arbitrary input image sizes without any resizing operation.

6.3 Experiments

We conduct experiments on handwritten digit strings obtained from CVL and ORAND-CAR datasets (provided by Diem et al. [26]) to demonstrate the effectiveness of our proposed CFSPP network, and compare its performance with that of all the participating methods of ICFHR2014 HDSR competition. To further verify the practicability of the proposed network and the priority of the FSPP layer, we also carry out experiments on our own cell-phone number dataset named PhPAIS.

6.3.1 Datasets

CVL is collected amongst 300 students from the Vienna University of Technology. Images in this dataset are with no background noise, but texts presented in these images are written with different colours.

ORAND-CAR is split into CAR-A and CAR-B since it is obtained from real cheques provided by two different banks. Notably, text images from CAR-A and CAR-B are with different background patterns and cheque layouts.

PhPAIS is our private dataset collected from the real China post mail images. As shown in Fig. 6.6, we manually crop and label handwritten phone numbers from these images, and finally, 11477 handwritten text images are collected. In addition, from the comparison with CVL and ORAND-CAR in Fig. 6.7, we can see that, strings presented in PhPAIS are much longer, and thus the recognition task on PhPAIS is much more challenging.

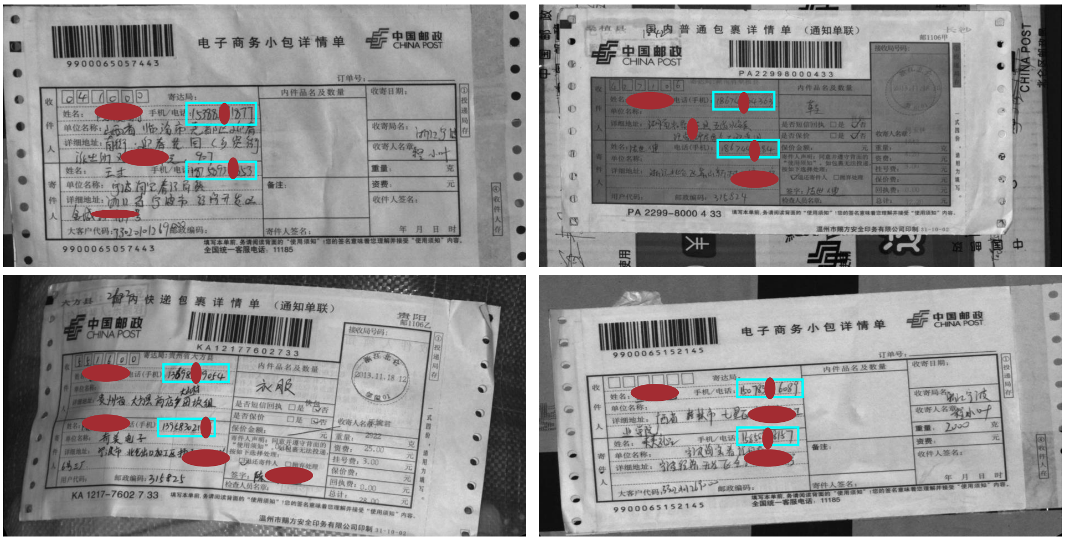


Figure 6.6 : China post mail images used to collect handwritten phone numbers for PhPAIS dataset. Some contents are covered by masks for privacy protection.

Fig. 6.7 exhibits some samples from the datasets described above. Some digits of strings from the PhPAIS dataset are covered for privacy protection. Apparently, compared with scene texts, handwritten texts are with various handwriting styles and suffer from connected characters. Distribution of the samples in each dataset with respect to string length is shown in Table 6.1.

6.3.2 Implementation Details

We utilize Rectified Linear Units (ReLU) as our activation function and optimize the proposed network via SGD with momentum. The learning rate is initialized to 0.01 and reduced by half per epoch. The precision is defined as the number of correctly recognized strings divided by the total number of strings, which is the same as the hard metric defined in ICFHR2014 competition [26]. Additionally, our

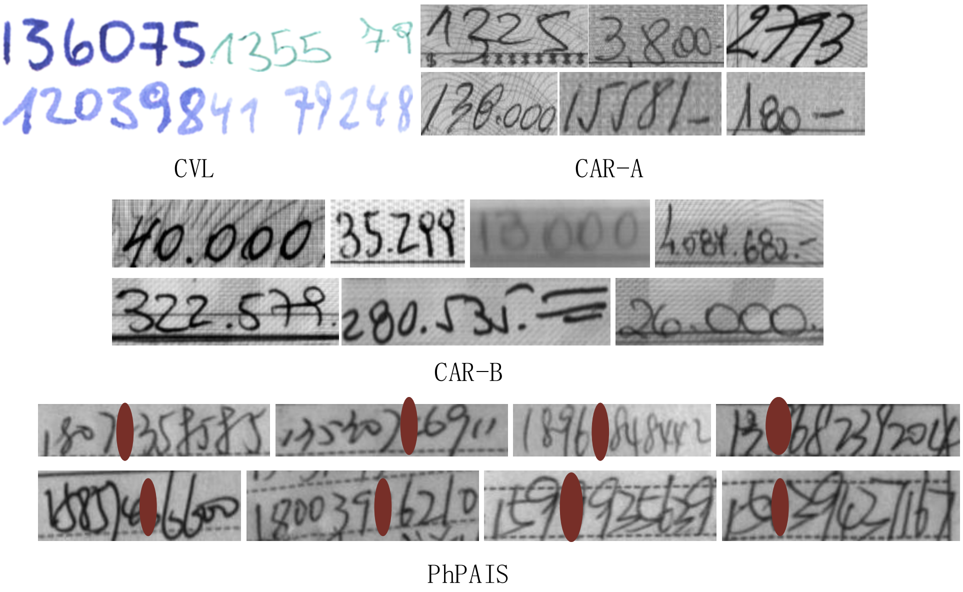


Figure 6.7 : Samples from CVL, ORAND-CAR and PhPAIS.

Table 6.1 : Distribution of the four different databases with respect to string length

Len	train				test			
	CVL	CAR-A	CAR-B	PhPAIS	CVL	CAR-A	CAR-B	PhPAIS
2	0	22	0	0	0	36	0	0
3	0	204	0	0	0	387	5	0
4	0	704	63	0	0	1425	69	0
5	125	903	1200	0	789	1475	1241	0
6	758	145	1599	828	4144	363	1452	912
7	379	29	137	244	1765	87	157	350
8	0	2	1	264	0	11	2	379
9	0	0	0	178	0	0	0	313
10	0	0	0	364	0	0	0	586
11	0	0	0	3122	0	0	0	3937
Total	1262	2009	3000	5000	6698	3784	2926	6477

network is coded in Matlab, C++ and CUDA languages without using any other open frameworks such as *caffe*, *torch*, *tensorflow*, etc.

6.3.3 Training Strategies

From Table 6.1, we can see that the length of digit strings ranges from 2 to 11. However, it does not mean all of the digits are totally connected (as shown in Fig. 6.6). Actually, handwritten digit strings usually consist of single digits and connect patterns with shorter length, and it is not hard to split these digit strings. Therefore, our network only needs to deal with the single digits and the connected patterns. According to our statistics, the dominant lengths of connected patterns in CVL and CAR datasets are 1, 2 and 3, so it is reasonable to set the maximal input sequence length $K = 3$ for the CVL and CAR datasets. The connection situation of PhPAIS is more serious, so we set $K = 5$ for this dataset.

To train the proposed network, we generate new datasets croppedCVL, croppedCAR-A, croppedCAR-B and croppedPhPAIS by manually cropping the digit images from CVL, CAR-A, CAR-B and PhPAIS. The new datasets are composed of single digits and strings with 1 to 5 digits. Corresponding distribution is presented in Table 6.2.

Table 6.2 : Distribution of the cropped datasets with respect to string length

Len	croppedCVL		croppedCAR-A		croppedCAR-B		croppedPhPAIS	
	train	test	train	test	train	test	train	test
1	7613	4714	8985	3667	15515	4386	48248	7133
2	6434	3977	6957	2886	13198	3599	42934	6368
3	5267	3206	4985	2088	10489	2802	37710	5630
4	0	0	0	0	0	0	33318	5211
5	0	0	0	0	0	0	28207	4425
total	19314	11897	20927	8641	39202	10787	190417	28767

Recognition accuracies of the proposed network with respect to different string length on croppedCVL, croppedCAR-A and croppedCAR-B datasets are shown in Table 6.3. Since the connection situation of these datasets are not very serious, we just use SPP layer in this network and set the scale to 4 in this layer. As seen, though the background of ORAND-CAR is very complicated, our proposed network is still capable of achieving the accuracies higher than 95% on both croppedCAR-A and croppedCAR-B datasets. We attribute the success to the deep and properly designed network structure and to the huge amount of training samples. Additionally, though CAR-B is the most challenging dataset among CVL, CAR-A and CAR-B, our network achieves the best performance on croppedCAR-B dataset as shown in Table 6.3, especially for strings with length of three. The reason is implied in Table 6.3, where the amount of training samples in croppedCAR-B is about two times of that in croppedCVL or croppedCAR-A. Therefore, we can figure out that more samples are very helpful for improving the performance of the proposed network, and it is very likely to improve the accuracies on croppedCVL and croppedCAR-A by providing more training samples.

Table 6.3 : Recognition accuracies (%) on the cropped datasets

Len	1	2	3	Mean
croppedCVL	96.37	89.21	78.32	89.11
croppedCAR-A	98.06	95.29	89.61	95.09
croppedCAR-B	98.72	95.97	92.65	96.23

6.3.4 Performance of the Proposed Network

We compare our method with the submissions of ICFHR2014 HDSR competition under the same evaluation metrics. To recognize handwritten strings in CVL and ORAND-CAR, we coarsely segment original images into single digits and connected patterns by using horizontal projection and connected component analysis. Then, the trained sequence labelling network is used to recognize each segment and the final results are obtained by simply joining the recognition results of all segments together.

Comparison results are shown in Table 6.4. Obviously, the proposed method achieves the highest mean accuracy on the three datasets. Especially, the accuracies on both challenging CAR-A and CAR-B datasets outperform the other methods by a large margin. According to [26], Beijing along with Pernambuco outperformed the other participating methods. The mean accuracies on CAR-A and CAR-B datasets of the Beijing and Pernambuco are (80.73% + 70.13%)/2 = 75.43% and (78.30% + 75.43%)/2 = 76.87%, respectively. Our method gets a mean accuracy of (82.61% + 83.32%)/2 = 82.97%, which is 7.54% and 6.10% higher than the Beijing and Pernambuco methods, respectively. Beijing is a well-designed over-segmentation method and Pernambuco is a combination of multiple classifiers including a hybrid classifier combining k-NN and SVM, and a hybrid classifier combining SVM and MDRNN [37]. The combination of k-NN and SVM is supposed to be an over-segmentation method while the combination of SVM and MDRNN is a segmentation-free method. Besides that, Pernambuco takes real data extracted from Brazilian bank cheques of last decades into account. Singapore combines HOG feature and RNN to handle this task, but due to the small amount of training samples, it doesn't perform well.

Table 6.4 : Recognition accuracies (%) of different methods

Method	CAR-A	CAR-B	CVL	Mean	Framework
Tebessa I	37.05	26.62	59.30	40.99	over-segmentation
Tebessa II	39.72	27.72	61.23	42.89	over-segmentation
Singapore	52.30	59.60	50.40	54.10	segmentation free
Pernambuco	78.30	75.43	58.60	70.78	combined both
Beijing	80.73	70.13	**85.29**	78.72	over-segmentation
Proposed	**82.61**	**83.32**	79.23	**81.72**	—-

From Fig. 6.5, we can see that both CAR-A and CAR-B are disturbed by background clutter and suffer from character connection problem. It seems there is no difference between these two datasets. However, according to Table 6.3, the dominant length of CAR-A is four and five, while it is five and six for CAR-B. The little difference has caused a large margin in recognition accuracy. As shown in Table 6.4, all the methods achieved higher accuracies in CAR-A than CAR-B except Singapore. Apparently, these methods are very sensitive to string length. However, our method achieves very similar accuracies on the two datasets, so the

Table 6.5 : Recognition accuracies (%) on the croppedPhPAIS and PhPAIS datasets

		croppedPhPAIS					PhPAIS
Set-ting	Pooling &scale	Len=1	Len=2	Len=3	Len=4	Len=5	
1	SPP&4	99.45	98.79	97.10	95.30	91.82	84.47
2	SPP&3	99.36	98.38	97.26	94.53	89.99	82.49
3	FSPP &2,3,4	99.44	98.45	97.19	94.59	91.77	84.31
4	FSPP &3,4,5	99.48	98.74	97.39	95.16	92.45	84.72

proposed method is more robust to string length. We blame the low accuracy of our method on CVL to the lack of diversity of this dataset since only 26 different strings are included by its 7960 samples (only 1262 for training). Though it is 6% lower than Beijing on CVL dataset, our method still achieves an accuracy that is 20% higher than Pernanmbuco. Overall, our method outperforms all of the participating methods of ICFHR2014 competition with a mean accuracy of 81.72% on the three datasets.

6.3.5 Comparison of SPP and FSPP on PhPAIS Dataset

The superiority of the proposed FSPP is evaluated on the PhPAIS dataset. Images in this dataset suffer from much longer string length (mostly 11) and more serious connection situation (maximal string length of connected patterns is up to 5). The proposed network is evaluated under four settings, which share the same configuration except the SPP/FSPP layer. Details and recognition results are shown in Table 6.5. Aspect ratio intervals of FSPP in setting 3 and 4 are $(0, 0.5)$, $[0.5, 0.9)$ and $[0.9, \infty)$. Network trained on the croppedPhPAIS is used to recognize the full-length strings of the PhPAIS dataset by cooperating with some simple coarse segmentation strategies.

Comparing recognition results of the first two settings, we can figure out that, reducing scales of the SPP layer will not cause significant performance degradation to the short strings, which means that too many features is redundant and unnecessary for short strings. In contrast, a larger performance degradation can be observed on longer strings. Intuitively, longer strings need richer features to discriminate characters at each position, so increasing scales is preferable to reducing them. Overall, scale reduction leads to a performance degradation of 2% on the PhPAIS dataset. When the more flexible FSPP is employed, different scales are set according to aspect ratios that related to string lengths. Obviously, the performance degradation is alleviated in a large degree and only 0.16% accuracy reduction is caused. Furthermore, we can reduce the scales for short strings and increase the scales for long strings at the same time by using FSPP as the fourth setting does. By this means, performance is slightly improved when comparing the first and the last settings. In conclusion, the proposed FSPP is a better choice for strings with longer string lengths and a more serious connection situation.

6.4 Conclusion

This work has presented a sequence labelling convolutional neural network for handwritten text recognition. We have reimplemented the SPP layer to handle the arbitrary string length problem, and designed an output layer with multiple softmax classifiers to deal with sequence labelling issue. Additionally, we have proposed a more flexible pooling strategy called FSPP on the basis of SPP to promote the performance of the proposed network on long strings. The proposed network directly recognize handwritten strings without segmenting them, so the challenges posed by connected patterns are avoided. Experimental results have shown that by combining with some simple coarse segmentation strategies, the proposed network is able to process long handwritten strings.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Texts are priceless treasure inherited from our ancestors, and for thousands of years, they have been playing critical roles in promoting human civilization, especially in the current age of information explosion. Reading texts, including scene texts and handwritten texts, from images is becoming more and more important with the dramatic development of techniques on image capturing, especially those for mobile devices.

From our research, we can see that text detection and recognition have entered a deep learning era. Deep network-based text detectors and recognisers have surpassed traditional ones by large margins in both structures and performances. At present, the most state-of-the-art approaches are on the base of CNN and LSTM. In this work, we have studied drawbacks of these approaches and proposed the solutions as follows.

1. To improve the recall rate of small text areas in oriented text detection, we propose to insert multiple ASPP modules into Xception after feature maps with different resolutions. This strategy significantly improves the recall rate of small text areas and F-measure. We also introduce auxiliary connections and auxiliary losses to speed up convergence and boost the discrimination ability of lower encoder layers.

2. To address the issue that the RoIAlign module of Mask R-CNN cannot fully leverage global information when performing predictions, we design a global mask module to perform semantic segmentation while considering global information and enhance features extracted by FPN with the predicted results.

The proposed module is trained in a supervised way and is able to effectively improve detection performance.

3. To tackle the problem that LSTM neglects the valuable spatial and structural information of 2-D text images, we propose a scene text recogniser that exploits ConvLSTM to directly perform sequential transcription in a 2-D space, and a scene text recogniser that utilizes one-hot encoded locations to enhance features with pixels' spatial information. Both recognisers achieve promising performances on recognizing low-resolution texts, noisy texts and curved texts.

4. To solve the problem that CNNs with fully connected layers are not suitable for sequential prediction tasks, we propose a CNN-based sequential labelling network and apply it to handwritten text recognition. The proposed network embeds an SPP-based intermediate layer between convolutional layers and fully connected layers to convert arbitrary-size feature maps into feature vectors with specific lengths, which are decided adaptively according to the aspect ratios of input text images. The proposed network is able to effectively recognize handwritten digit strings on real-word bank cheques and handwritten phone numbers.

7.2 Future Works

CNN and LSTM play critical roles in text detection and recognition. Thought detectors and recognisers applying CNN and LSTM have achieved promising performance, there are still many other interesting and challenging topics about text reading, as presented below.

1. Generating synthetic handwritten text images. A large number of scene text images have been produced via deep networks to improve the performance of scene text detection and recognition, but techniques for handwritten text generation have not been studied.

2. Improving performance with unlabelled data. Capturing text images is easy,

but annotating them is exhausted and time-consuming. Therefore, utilizing unlabelled or partially labelled data for text detection and text recognition with unsupervised learning or semi-supervised learning methods is important.

3. Scene text visual question answering. Texts presenting in images convey important semantic information and help us to understand the man-made environments better. Therefore, understanding images while considering texts is attracting more and more attentions from the community of computer vision.

4. Searching better network architectures with Neural Network Search (NAS) techniques. According to literature, networks searched by NAS have achieved better efficiency and effectiveness than manually designed networks. Therefore, seeking more powerful deep networks via NAS for text detection and text recognition is supposed to be helpful.

5. Exploring solutions to data scarcity. Data scarcity is a common problem in computer vision tasks, especially for text detection and text recognition. Domain adaption and one/few-shot learning are the two most effective solutions. Therefore, in our future work, we will also conduct researches in this direction.

Bibliography

[1] L. V. Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, “recaptcha: human-based character recognition via web security measures,” *Science*, vol. 321, pp. 1465–1468, 2008.

[2] G. Alex, “Supervised sequence labelling with recurrent neural networks,” *Springer*, 2012.

[3] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in *ICLR*, 2015.

[4] B. Bai, F. Yin, and C.-L. Liu, “Scene text localization using gradient local correlation,” in *12th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2013, pp. 1380–1384.

[5] F. Bai, Z. Cheng, Y. Niu, S. Pu, and S. Zhou, “Edit probability for scene text recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[6] X. Bai, M. Yang, B. Shi, and L. Minghui, “Deep learning for scene text detection and recognition,” *SCIENTIA SINICA Informationis*, vol. 48, pp. 531–544, 2018.

[7] C. Bartz, H. Yang, and C. Meinel, “Stn-ocr: a single neural network for text detection and recognition,” *CoRR*, vol. arXiv preprint arXiv: 1707.08831v1, 2017.

[8] T. Bluche, J. Louradour, and R. Messina, “Scan, attend and read: end-to-end handwritten paragraph recognition with mdlstm attention,” *CoRR*, vol. arXiv preprint arXiv: 1604.03286, 2016.

[9] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean, “Large language models in machine translation,” in *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2007, pp. 858–867.

[10] H. Chen, S. S. Tsai, G. Schroth *et al.*, “Robust text detection in natural images with edge-enhanced maximally stable extrmal regions,” in *IEEE International Conference on Image Processing*, 2011, pp. 2609–2612.

[11] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Semantic image segmentation with deep convolutional nets and fully connected CRFs,” in *ICLR*, 2015.

[12] ——, “Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution and fully connected CRFs,” *IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 40, pp. 834–848, 2017.

[13] L.-C. Chen, G. Papandreous, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” *CoRR*, vol. arXiv preprint arXiv: 1706.05587, 2017.

[14] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *European Conference on Computer Vision (ECCV)*, 2018.

[15] Z. Cheng, F. Bai, Y. Xu, G. Zheng *et al.*, “Focusing attention: towards accurate text recognition in natural images,” in *International Conference on Computer Vision (ICCV)*, 2017, pp. 5086–5094.

[16] Z. Cheng, Y. Xu, F. Bai, Y. Niu, S. Pu, and Z. Shuigeng, “AON: arbitrarily-oriented text recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[17] Z. Cheng, Y. Xu, F. Bai, Y. Niu, S. Pu, and S. Zhou, “Aon: towards arbitrarily-oriented text recognition,” in *International Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5571–5579.

[18] C. K. Chng and C. S. Chan, “Total-text: a comprehensive dataset for scene text detection and recognition,” *CoRR*, vol. arXiv preprint arXiv: 1710.10400, 2017.

[19] K. Cho, B. V. Merrinboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: encoder-decoder approaches,” *CoRR*, vol. arXiv preprint arXiv: 1409.1259, 2014.

[20] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Chou, and Y. Bengio, “Attention-based models for speech recognition,” *CoRR*, vol. arXiv preprint arXiv: 1506.07503, 2015.

[21] H. Choudhury and S. M. Prasanna, “Handwriting recognition using sinusoidal model parameters,” *Pattern Recognition Letter (PRL)*, vol. 000, pp. 1–10, 2018.

[22] J. Dai, H. Qi, Y. Xiong, Y. Li *et al.*, “Deformable convolutional networks,” *CoRR*, vol. arXiv preprint arXiv: 1703.06211, 2017.

[23] Y. Dai, Z. Huang, Y. Gao *et al.*, “Fused text segmentation networks for multi-oriented scene text detection,” in *International Conference on Pattern Recognition (ICPR)*, 2018, pp. 3604–3609.

[24] D. Deng, H. Liu, X. Li, and D. Cai, “Pixellink: detecting scene text via instance segmentation,” in *AAAI*, 2018.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-f. Li, “Imagenet: a large-scale hierarchical image database,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[26] M. Diem, S. Fiel, F. Kleber, R. Sablatnig *et al.*, “ICFHR 2014 competition on handwritten digit string recognition in challenging datasets (hdsrc 2014),” in *14th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2014.

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

[28] M. Everingham, L. V. Gool, C. K. Williams, J. Winn, and A. Zisserman, “The PASCAL visual object classes (VOC) challenge,” *International Journal of Computer Vision*, vol. 88, pp. 303–338, 2010.

[29] S. Gao, C. Wang, B. Xiao, C. Shi, Y. Zhang, Z. Lv, and S. Yanqin, “Adaptive scene text detection based on transferring adaboost,” in *12th International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 388–392.

[30] S. Gao, C. Wang, B. Xiao, C. Shi, W. Zhou, and Z. Zhang, “Scene text recognition by learning co-occurrence of strokes based on spatially embedded dictionary,” *IET Computer Vision*, vol. 9, no. 1, pp. 138–148, 2015.

[31] Y. Gao, Y. Chen, J. Wang, and H. Lu, “Reading scene text with attention convolutional sequence modeling,” *CoRR*, vol. arXiv preprint arXiv: 1709.04303v1, 2017.

[32] Y. Gao, Y. Chen, J. Wang, M. Tang, and H. Lu, “Dense chained attention network for scene text recognition,” in *International Conference on Image Processing (ICIP)*, 2018, pp. 679–683.

[33] R. Girshick, “Fast R-CNN,” in *International Conference on Computer Vision (ICCV)*, 2015.

[34] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, “Multi-digit number recognition from street view imagery using deep convolutional neural networks,” *CoRR*, vol. arXiv preprint arXiv: 1312.6082v4, 2014.

[35] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labeling unsegmented sequence data with recurrent neural networks,” in *International Conference on Machine Learning*, 2006, pp. 369–376.

[36] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, “A novel connectionist system for unconstrained handwriting recognition,” *IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 31, no. 5, pp. 855–868, 2009.

[37] A. Graves and J. Schmidhuber, “Offline handwriting recognition with multidimensional recurrent neural networks,” in *NIPS*, 2008.

[38] A. Gupta, A. Vedaldi, and A. Zisserman, “Synthetic data for text localization in natural images,” in *International Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2315–2324.

[39] D. He, X. Yang, C. Liang, Z. Zhou *et al.*, “Multi-scale FCN with cascaded instance aware segmentation for arbitrary oriented word spotting in the wild,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[40] K. He, G. Gkioxari, P. Dollar, and R. Crishick, “Mask r-cnn,” *CoRR*, vol. arXiv preprint arXiv: 1703.06211, 2017.

[41] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 37, no. 9, pp. 1904–1916, 2015.

[42] ——, “Deep residual learning for image recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[43] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, “Single shot text detector with regional attention,” in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[44] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, “Deep direct regression for multi-oriented scene text detection,” in *International Conference on Computer Vision (ICCV)*, 2017, pp. 745–753.

[45] S. Hochreiter and J. Schmihuber, “Long short-term memory,” *Neural Computation*, vol. 9, pp. 1735–1780, 1997.

[46] Z. Huang, Z. Zhong, L. Sun, and Q. Huo, “Mask r-cnn with pyramid attention network for scene text detection,” in *IEEE Winter Conference on Application of Computer Vision*, 2019, pp. 764–772.

[47] S. Ioffe and C. Szegedy, “Batch normalization: accelerating deep network training by reducing internal covariate shift,” *CoRR*, vol. arXiv preprint arXiv: 1502.03167v3, 2015.

[48] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, “Synthetic data and artificial neural networks for natural scene text recognition,” *CoRR*, vol. arXiv preprint arxiv:1412.1842, 2014.

[49] Y. Jiang, X. Zhu, X. Wang, S. Yang *et al.*, “R^2CNN: rotational region cnn for orientation robust scene text detection,” *CoRR*, vol. arXiv preprint arXiv: 1706.09579, 2017.

[50] C. John, “A computational approach to edge detection,” *IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 8, pp. 679–714, 1986.

[51] T. Judd, K. Ehinger, F. Durand, and A. Torralba, “Learning to predict where humans look,” in *IEEE Conference on Computer Vision(ICCV)*, 2009, pp. 2106–2113.

[52] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh *et al.*, “ICDAR 2015 robust reading competition,” in *13th International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 1156–1160.

[53] D. Karatzas, F. Shafait, S. Uchida *et al.*, “ICDAR 2013 robust reading competition,” in *12th International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 1484–1493.

[54] D. Keysers, T. Deselaers, H. A. Rowley, L.-L. Wang, and V. Carbune, “Multi-language online handwriting recognition,” *IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 39, no. 6, pp. 1180–1194, 2017.

[55] K.-H. Kim, S. Hong, B. Roh, Y. Cheon, and M. Park, “PVANET: deep but lightweight neural networks for real-time object detection,” *CoRR*, vol. arXiv preprint arXiv: 1608.08021, 2016.

[56] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *IEEE Proceedings*, vol. 86, pp. 2278–2324, 1998.

[57] C.-Y. Lee and S. Osindero, “Recursive recurrent nets with attention modeling for ocr in the wild,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2231–2239.

[58] H. Lee and B. Verma, “Binary segmentation algorithm for english cursive handwriting recognition,” *Pattern Recognition (PR)*, vol. 45, pp. 1306–1317, 2012.

[59] H. Li, P. Wang, C. Shen, and G. Zhang, “Show, attend and read: a simple and strong baseline for irregular text recognition,” in *AAAI*, 2019.

[60] X. Li, W. Wang, W. Hou, R.-Z. Liu, T. Lu, and J. Yang, “Shape robust text detection with progressive scale expansion network,” *CoRR*, vol. arXiv preprint arxiv:1806.02559, 2018.

[61] Y. Li, Y. Yu, Z. Li, Y. Lin, M. Xu, J. Li, and X. Zhou, “Pixel-anchor: a fast oriented scene text detector with combined network,” *CoRR*, vol. arXiv preprint arXiv: 1811.07432v1, 2018.

[62] Z. Li, K. Gavrilyuk, E. Gavves, M. Jain, and C. G. Snoek, “Videolstm convolves, attends and flows for action recognition,” *Computer Vision and Image Understanding*, vol. 166, pp. 41–50, 2018.

[63] M. Liao, B. Shi, and X. Bai, “Textboxes++: a single shor oriented scene text detector,” *IEEE Transaction on Image Processing*, vol. 27, pp. 3676–3690, 2018.

[64] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, “Textboxes: a fast text detector with a single deep neural network,” in *AAAI*, 2017, pp. 4161–4167.

[65] M. Liao, J. Zhang, Z. Wan, F. Xie, J. Liang *et al.*, “Scene text recognition from two-dimensional perspective,” in *AAAI*, 2019.

[66] M. Liao, Z. Zhu, B. Shi, G.-S. Xia, and X. Bai, “Rotation-sensitive regression for oriented scene text detection,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5909–5918.

[67] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[68] C.-L. Liu, M. Koga, and H. Fujisawa, “Lexicon-driven segmentation and recognition of handwritten character strings for Japanese address reading,” *IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 24, no. 11, pp. 1425–1437, 2002.

[69] C.-L. Liu, F. Yin, Q.-F. Wang, and D.-H. Wang, “ICDAR2011-Chinese handwriting recognition competition,” in *2011 International Conference on Document Analysis and Recognition (ICDAR)*, 2011, pp. 1464–1469.

[70] J. Liu, X. Liu, J. Sheng, D. Liang, X. Li, and Q. Liu, “Pyramid mask text detector,” in *International Conference on Computer Vision and Pattern Recognition*, 2019.

[71] ——, “Pyramid mask text detector,” in *International Conference on Computer Vision and Pattern Recognition*, 2019.

[72] M. Liu and M. Zhu, “Mobile video object detection with temporally-aware feature maps,” in *International Conference on Computer Vision and Pattern*

Recognition, 2018, pp. 5571–5579.

[73] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed *et al.*, “SSD: single shot multibox detector,” in *European Conference on Computer Vision (ECCV)*, 2016, pp. 21–37.

[74] W. Liu, C. Chen, and K.-Y. K. Wong, “Char-net: a character aware neural network for distorted scene text recognition,” in *AAAI*, 2018.

[75] W. Liu, C. Chen, K.-Y. K. Wong, Z. Su, and J. Han, “START-NET: a spatial attention residue network for scene text recognition,” *BMVC*, vol. 2, no. 7, pp. 1–13, 2016.

[76] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan, “Fots: fast oriented text spotting with a unified network,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5676–5685.

[77] Y. Liu, L. Jin, S. Zhang, and S. Zhang, “Detecting curve text in the wild: new dataset and new solution,” *CoRR*, vol. arXiv preprint arXiv: 1712.02170, 2017.

[78] Z. Liu, Y. Li, F. Ren, W. L. Goh, and H. Yu, “Squeezedtext: a real-time scene text recognition by binary convolutional encoder-decoder network,” in *AAAI*, 2018.

[79] M. Liwicki, G. Alex, H. Bunke, and J. Schmidhuber, “A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks,” in *12th International Conference on Document Analysis and Recognition (ICDAR)*, 2007, pp. 367–371.

[80] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.

[81] S. Long, X. He, and C. Yao, “Scene text detection and recognition: the deep learning era,” *CoRR*, vol. arXiv preprint arXiv: 1811.04256v3, 2018.

[82] S. Long, J. Ruan, W. Zhang, X. He *et al.*, "Textsnake: a flexible representation for detecting text of arbitrary shapes," in *European Conference on Computer Vision (ECCV)*, 2018.

[83] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai, "Mask textspotter: an end-to-end trainable neural network for spotting text with arbitrary shapes," in *European Conference on Computer Vision (ECCV)*, 2018.

[84] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented scene text detection via corner localization and region segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7553–7563.

[85] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue, "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Transaction on Multimedia*, 2018.

[86] U.-V. Marti and B. H., "A full English sentence database for off-line handwriting recognition," in *5th International Conference on Document Analysis and Recognition (ICDAR)*, 1999, pp. 705–708.

[87] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *the British Machine Vision Conference (BMVC)*, 2002, pp. 384–393.

[88] R. Messina and J. Louradour, "Segmentation-free handwritten Chinese text recognition with lstm-rnn," in *13th International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 171–175.

[89] A. Mishra, K. Alahari, and C. Jawahar, "Top-down and bottom-up cues for scene text recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2687–2694.

[90] ——, "Top-down and bottom-up cues for scene text recognition," in *International Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2687–2694.

[91] N. Nayef, F. Yin, I. Bizid *et al.*, "ICDAR2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt," in *2017 14th International Conference on Document Analysis and Recognition (ICDAR)*, 2017, pp. 1454–1459.

[92] L. Neumann and J. Matas, "Scene text localization and recognition with oriented stroke detection," in *International Conference on Computer Vision (ICCV)*, 2013, pp. 97–104.

[93] ——, "Text localization in real-world umages using efficiently pruned exhaustive search," in *2011 International Conference on Document Analysis and Recognition (ICDAR)*, 2011, pp. 687–691.

[94] ——, "Real-time scene text localization and recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[95] S. Nomura, K. Yamanaka, O. Karai *et al.*, "A novel adaptive morphological approach for degraded character image segmentation," *Pattern Recognition (PR)*, vol. 38, pp. 1961–1975, 2005.

[96] T. Q. Phan, P. Shivakumara, B. Su, and C. L. Tan, "A gradient vector flow-based method for video character segmentation," in *International Conference on Document Analysis and Recognition (ICDAR)*, 2011, pp. 1024–1028.

[97] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[98] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *NIPS*, 2015.

[99] A. Risnumawan, P. Shivakumara, C. S. Chan, and C. L. Tan, "A robust arbitrary text detection system for natural scene images," *Expert Systems with Applications*, vol. 41, pp. 8027–8048, 2014.

[100] X. Rong, C. Yi, and Y. Tian, "Unambiguous text localization and retrieval for cluttered scenes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3279–3287.

[101] J. A. Sanchez, V. Romero, A. H. Toselli, and E. Vidal, "ICFHR2016 competition on handwritten text recognition on the READ dataset," in *15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2016, pp. 630–635.

[102] J. A. Sanchez, V. Romero, A. H. Toselli, M. Villegas, and E. Vidal, "ICDAR2017 competition on handwritten text recognition on the READ dataset," in *14th International Conference on Document Analysis and Recognition (ICDAR)*, 2017, pp. 1383–1388.

[103] J. A. Sanchez, A. H. Toselli, V. Romero, and E. Vidal, "ICDAR 2015 competition HTRtS:handwritten text recognition on the transcriptorium dataset," in *13th International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 1166–1170.

[104] J. Schenk and G. Rigoll, "Novel hybrid NN/HMM modelling techniques for on-line handwriting recognition," in *International Workshop Frontiers Handwriting Recognition (ICPR)*, 2006, pp. 619–623.

[105] K. Sheshadri and S. K. Divvala, "Exemplar driven character recognition in the wild," in *British Machine Vision Conference*, 2012.

[106] B. Shi, X. Bai, and B. Serge, "Detecting oriented text in natural images by linking segments," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[107] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 39, no. 11, pp. 2298–2304, 2017.

[108] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4168–4176.

[109] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, "Aster: an attention scene text recognizer with flexible rectification," *IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 31, no. 11, pp. 855–868, 2018.

[110] C. Shi, C. Wang, B. Xiao, Y. Zhang, and S. Gao, "Scene text detection using graph model built upon maximally stable extemal regions," *Pattern Recognition Letter*, vol. 34, pp. 107–116, 2013.

[111] C. Shi, C. Wang, B. Xiao, Y. Zhang, S. Gao, and Z. Zhang, "Scene text recognition using part-based tree-structured character detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 2961–2968.

[112] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo, "Convolutional lstm network: a machine learning approach for precipitation nowcasting," in *NIPS*, 2015.

[113] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. arXiv preprint arXiv: 1409.1556, 2014.

[114] T. StrauB, G. Leifert, R. Labahn *et al.*, "ICFHR2018 competition on automated text recognition on a READ dataset," in *16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2018, pp. 477–482.

[115] T.-H. Su, T.-W. Zhang, D.-J. Guan, and H.-J. Huang, "Off-line recognition of realistic chinese handwriting using segmentation-free strategy," *Pattern Recognition (PR)*, vol. 42, no. 1, pp. 167–182, 2009.

[116] J. Sueiras, V. Ruiz, A. Sanchez, and J. F. Velez, "Offline continuous handwriting recognition using sequence to sequence neural networks," *Neurocomputing*, vol. 289, pp. 119–128, 2018.

[117] L. Sun, T. Su, C. Liu, and R. Wang, "Deep lstm networks for online Chinese handwriting recognition," in *15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2016, pp. 271–276.

[118] C. Szegedy, W. Liu, Y. Jia *et al.*, "Going deeper with convolutions," *CoRR*, vol. arXiv preprint arXiv: 1409.4842, 2014.

[119] C. Szegedy, V. Vanhoucke, S. Ioffe *et al.*, "Rethinking the inception architecture for computer vision," in *International Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[120] S. Tian, U. Bhattacharya, S. Lu, B. Su, Q. Wang *et al.*, "Multilingual scene character recognition with co-occurrence of histogram of oriented gradients," *Pattern Recognition (PR)*, vol. 51, pp. 125–134, 2016.

[121] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 56–72.

[122] D.-H. Wang, C. Liu, and X.-D. Zhou, "An approach for real-time recognition of online Chinese handwritten sentences," *Pattern Recognition (PR)*, vol. 45, pp. 3661–3675, 2012.

[123] F. Wang, L. Zhao, X. Li, X. Wang, and D. Tao, "Geometry-aware scene text detection with instance transform network," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1381–1389.

[124] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *IEEE International Conference on Computer Vision*, 2011, pp. 1457–1464.

[125] Q. Wang, W. Jia, X. He *et al.*, "Reelfa: a scene text recognizer with encoded location and focused attention," in *International Conference on Document Analysis and Recognition Workshops (ICDARW)*, 2019, pp. 71–76.

[126] Q. Wang, W. Jia, X. He, Y. Lu, M. Blumenstein *et al.*, "Deeptext: detecting text from the wild with multi-aspp-assembled DeepLab," in *International*

Conference on Document Analysis and Recognition (ICDAR), 2019.

[127] Q. Wang, W. Jia, X. He, Y. Lu, M. Blumenstein, and Y. Huang, "Faclstm: Convlstm with focused attention for scene text recognition," *CoRR*, vol. arXiv preprint arXiv: 1904.09405v1, 2019.

[128] Q. Wang and Y. Lu, "A sequence labeling convolutional network and its application to handwritten string recognition," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2017, pp. 2950–2956.

[129] Q.-F. Wang, F. Yin, and C.-L. Liu, "Integrating language model in handwritten Chinese text recognition," in *10th International Conference on Document Analysis and Recognition (ICDAR)*, 2009, pp. 1036–1040.

[130] ——, "Handwritten chinese text recognition by integrating multiple contexts," *IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 34, no. 8, pp. 1469–1481, 2012.

[131] T. Wang, D. J. Wu, A. Coates, and A. Y. NG, "End-to-end text recognition with convolutional neural networks," in *International Conference on Pattern Recognition (ICPR)*, 2012, pp. 3304–3308.

[132] Z.-R. Wang, J. Du, W.-C. Wang, J.-F. Zhai, and J.-S. Hu, "A comprehensive study of hybrid neural network hidden markov model for offline handwritten Chinese text recognition," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 21, pp. 241–251, 2018.

[133] Z. Wojna, A. Gorban, D.-S. Lee, K. Murphy, Q. Yu *et al.*, "Attention-based extraction of structured information from street view imagery," in *International Conference on Document Analysis and Recognition*, 2017, pp. 844–850.

[134] C. Wolf and J.-M. Jolion, "Object count/area graphs for the evaluation of object detection and segmentation algorithms," *International Journal of Document Analysis and Recognition (IJDAR)*, pp. 280–296, 2006.

[135] Y.-C. Wu, F. Yin, and C.-L. Liu, “Improving handwritten Chinese text recognition using neural network language models and convolution neural network shpe models,” *Pattern Recognition (PR)*, vol. 65, pp. 251–264, 2017.

[136] Y. Wu and P. Natatajan, “Self-organized text detection with minimal post-processing via border learning,” in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 5000–5009.

[137] E. Xie, Y. Zhang, S. Shao, G. Yu, C. Yao, and G. Li, “Scene text detection with supervised pyramid context network,” in *AAAI*, 2019.

[138] Z. Xie, Z. Sun, L. Jin, Z. Feng, and S. Zhang, “Fully convolutional recurrent network for handwritten Chinese text recognition,” in *International Conference on Pattern Recognition (ICPR)*, 2016, pp. 4011–4016.

[139] C. Xue, S. Lu, and Z. Fangneng, “Accurate scene text detection through border semantics awareness and bootstrapping,” in *European Conference on Computer Vision (ECCV)*, 2018.

[140] M. Yang, Y. Guan, M. Liao, X. He, K. Bian, S. Bai, C. Yao, and X. Bai, “Symmetry-constrained rectification network for scene text recognition,” in *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[141] Q. Yang, M. Cheng, W. Zhou, Y. Chen, M. Qiu, and W. Lin, “Inceptext: a new inception-text module with deformable psroi pooling for multi-oriented scene text detection,” in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018, pp. 1071–1077.

[142] X. Yang, D. He, Z. Zhou, D. Kifer, and C. L. Giles, “Learning to read irregular text with attention mechanisms,” in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2017, pp. 3280–3286.

[143] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, “Detecting text of arbitrary orientations in natural images,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1083–1090.

[144] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao, “Scene text detection via holistic, multi-channel prediction,” *CoRR*, vol. arXiv preprint arXiv: 1606.09002v2, 2016.

[145] Q. Ye and D. Doermann, “Text detection and recognition in imagery: A survey,” *IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 37, no. 7, pp. 1480–1500, 2015.

[146] F. Yin, Q.-F. Wang, X.-Y. Zhang, and C.-L. Liu, “ICDAR 2013 Chinese handwriting recognition competition,” in *12th International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 1095–1101.

[147] F. Yin, Y.-C. Wu, X.-Y. Zhang, and L. Cheng-Lin, “Scene text recognition with sliding convolutional character models,” *CoRR*, vol. arXiv preprint arXiv: 1709.01727, 2017.

[148] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, “Robust text detection in natural scene images,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 970–983, 2014.

[149] X. Yin, X. Yin, H. Hao, and K. Iqbal, “Effective text localization in natural scene images with MSER, geometry-based grouping and adaboost,” in *21st International Conference on Pattern Recognition (ICPR)*, 2012, pp. 725–728.

[150] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and H. Thomas, “Unitbox: an advanced object detection network,” in *ACM on Multimedia Conference*, 2016, pp. 516–520.

[151] L. Zhang, G. Zhu, L. Mei, P. Shen *et al.*, “Attention in convolutional lstm for gesture recognition,” in *NIPS*, 2018.

[152] S. Zhang, Y. Liu, L. Jin, and C. Luo, “Feature enhanced network: a refined scene text detector,” in *AAAI*, 2018.

[153] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, “Multi-oriented text detection with fully convolutional networks,” in *IEEE Conference on*

Computer Vision and Pattern Recognition, 2016, pp. 4159–4167.

[154] X.-D. Zhou, D.-H. Wang, F. Tian, C.-L. Liu, and M. Nakagawa, "Handwritten Chinese/Japanese text recognition using semi-markov conditional random fields," *IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 35, no. 10, pp. 2413–2426, 2013.

[155] X.-D. Zhou, Y.-M. Zhang, F. Tian, H.-A. Wang, and C.-L. Liu, "Minimum-risk training for semi-markov conditional random fields with application to handwritten Chinese/Japanese text recognition," *Pattern Recognition (PR)*, vol. 47, no. 5, pp. 1904–1916, 2014.

[156] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "East: an efficient and accurate scene text detector," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5551–5560.

[157] G. Zhu, L. Zhang, P. Shen, and J. Song, "Multimodal gesture recognition using 3-d convolution and convolutional lstm," *IEEE Access*, vol. 5, pp. 4517–4524, 2017.

[158] X. Zhu, Y. Jiang, S. Yang, X. Wang, W. Li, P. Fu, H. Wang, and Z. Luo, "Deep residual text detection network for scene text," in *12th International Conference on Document Analysis and Recognition (ICDAR)*, 2017, pp. 807–812.

[159] Y. Zhu, C. Yao, and X. Bai, "Scene text detection and recognition: recent advances and future trends," *Frontiers of computer science*, vol. 10, no. 1, pp. 19–36, 2016.