SDINet Scheme for Generalized Text Detection in Scene and Document Images

by

Pravir Pal
202011044

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY
in
INFORMATION AND COMMUNICATION TECHNOLOGY
to

Dhirubhai Ambani Institute of Information and Communication Technology

May, 2022

Declaration

I hereby declare that

i) the thesis comprises of my original work towards the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,

ii) due acknowledgment has been made in the text to all the reference material used.

Pravir Pal

Pravir Pal

Certificate

This is to certify that the thesis work entitled SDINet Scheme for Generalized Text Detection in Scene and Document Images has been carried out by Pravir Pal for the degree of Master of Technology in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under our supervision.

Joshi 25/8/22

M.V. Joshi
Thesis Supervisor

Shruti

Shruti Bhilare
Thesis Co-Supervisor

Acknowledgments

I would like to thank my supervisor, Prof. M.V. Joshi and co-supervisor, Prof. Shruti Bhilare, for their continuous support and guidance in performing my research work. Besides my advisors, I would like to thank my sister, Punita Pal, and my friend, Shilpa Kayastha, for their constant emotional support and motivation to complete my thesis work. I would like to thank my friends I met at DAIICT. Thank you for helping me when I had a problem, and thank you for making my M.Tech journey fun and exciting. Last but not least, I would like to thank my parents. Without them, I would have nothing.

Contents

Abstract

Text Detection is an essential intermediate step in optical character recognition (OCR). OCR applied in scene text images is helpful for applications such as traffic signs and vehicle number plate recognition. OCR applied in the document text images help digitise and analyse the documents. Hence, a robust text detection system is needed to detect the text exceptionally well given an arbitrary text image. In this work, we address text detection in images using our Scene or Document Image Network (SDINet). During the training of the model, a Weighted Loss (WL) is designed to better update the training parameters according to the input image type. A classification model is designed that helps us to find the WL by classifying an input image as a scene text type image or document text type image. The novelty of our approach is in the fact that the training parameters of the model are updated according to the input image type. Our approach shows comparative results in all the evaluation parameters for scene text and document text datasets. Specifically, when compared to PSENet [27], experimental results show that our SDINet approach improves the recall by more than 1%, and F-score is increased by approximately 1% for SCUT-CTW 1500 dataset. [30].

List of Acronyms

BFS	Breadth-First-Search
BN	Batch Normalization
CCA	Connected Component Analysis
CNN	Convolutional Neural Networks
CRNN	Convolutional recurrent neural network
EAST	Efficient and Accurate scene text detector
F	F-score
IOU	Intersection Over Union
MSER	Maximally Stable Extremal Regions
NMS	Non-maximum suppression
OCR	Optical character recognition
PSENet	Progressive Scale Expansion Network
P	Precision
RELU	Rectified Linear Unit
R	Recall
SSD	Single shot multibox detector
SWT	Stroke Width Transform
WL	Weighted Loss
YOLO	You look only once

List of Tables

List of Figures

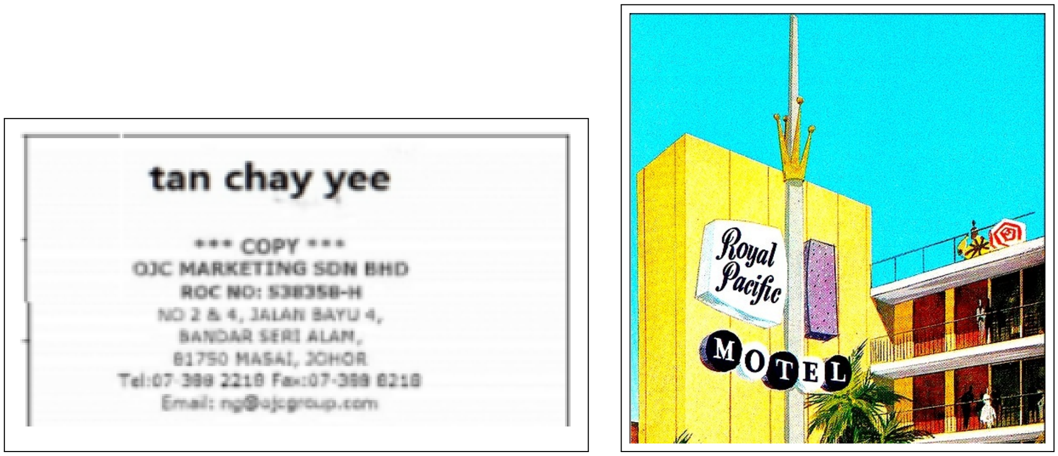
CHAPTER 1
Introduction

Detecting text in images can be crucial in extracting data from them. Although humans find the task extremely straightforward, identifying traits that define text can be difficult for machine learning algorithms. Text on some kind of congested objects or object structures with properties similar to texts in an image may make solving the challenge more difficult. Other problems in text detection are the arbitrary shape of text instances that make the separation of text instances more complicated, distortions, uneven lightning, variable text size and fonts.

There are several phases involved in detecting and extracting text. For meaning to emerge, text must first be identified inside the image, then extracted, and finally recognised. We concentrate on the first step of the pipeline, which involves attempting to locate text-containing regions in an image.

1.1 Problem Statement and Motivation

In this work, our primary focus is on detecting text from scene images or document images using our proposed SDINet scheme. Outdoor image such as in Fig. 1.1(b) is termed as scene image. Detection of text in such an image is difficult because the texts are curved in shape and have complex background changes. An image such as in Fig. 1.1(a) is termed as document text image. Detection of text in document images can be difficult because texts are densely located; because of this, semantic information may get lost. The demand for textual knowledge extraction from many sources has increased dramatically in this era of digitisation. Fortunately, recent advancements in computer vision have enabled us to make significant progress in text detection and other document analysis tasks. Optical character recognition is a technique used in computer vision to translate text found in photographs or scanned documents into a machine-readable format that can be altered, searched, and utilised for further processing [8]. OCR in scene text images and document text images covers several critical applications. This can be



(a) (b)

Figure 1.1: Different types of images containing text: (a) Cropped document text image from SROIE 2019 [6] dataset; (b) Scene text image from SCUT-CTW 1500 [30] dataset.

because the text in scene images compared to document images have mostly contrasting properties. Text in scene images is highly sparse, multi-oriented, curved in shape, and has varying font sizes with complex background changes. In contrast, text in document images is highly dense, less oriented, and has a comparatively even font size. OCR in scene text images is used to license recognise license plate numbers which can be used for vehicle tracking, toll collection, etc [19]. OCR in document text images plays a critical role for many enterprises and institutions with thousands of documents to process, analyse, and change daily. For example, information can easily be retrieved from receipts, invoices etc., using OCR [6]. As text detection is an important intermediate step in the process of OCR, given an arbitrary text image, a robust text detection system is needed to detect the text extremely well.

1.2 Contribution

Various methods have been proposed for detecting text from images recently [31, 27, 14, 32, 15, 23]. These methods are also briefly discussed in Section 2.2. However, out of these methods, only the PSENet [27] model is a shape robust text detection model that has performed well compared to other text detection models in general. Hence we chose the PSENet [27] model to apply the proposed SDINet scheme on it for our work.

The contribution of this work is as follows:

- An SDINet scheme is designed to better update the training parameters of

PSENet [27] according to the input image type.

1.3 Thesis Organization

The Thesis is structured as follows. In Chapter 2, the literature review is presented. In Chapter 3, the PSENet model is explained. Chapter 4 discusses our proposed SDINet scheme. Chapter 5 contains a discussion of experimental results, evaluation protocol, and performance comparison of the proposed scheme with other methods. Chapter 6 gives the conclusion of our work.

1.4 Chapter Summary

In this chapter, we first understood the importance of text detection. Then, we looked at various input images that can be supplied to the text detection model. After that, the problem statement is defined. The Motivation behind the problem statement is discussed by presenting various enterprise applications related to scene text type images or document text type images. In the end, the entire flow of the thesis is organized. The next chapter presents a literature review that deals with various kinds of text detection techniques used before and during the deep learning era.

CHAPTER 2
Literature Review

This chapter focuses on the current advancements in text detection. We will discuss some traditional and deep learning-based text detection methods and recent advances in deep learning-based text detection.

2.1 Text Detection Methods Prior to Deep Learning Era

In this section, we will go over some of the text detection techniques that were proposed before the deep learning era. During this time, the focus of detection algorithm development is on feature design.

Most text detection approaches use Sliding Window (SW) [2], [13], [26] or Connected Component Analysis (CCA) [3], [5], [28], [29] based categorization. The methods based on Connected Component Analysis (CCA), represented in Fig. 2.1, to perform classification of text/non-text on retrieved nominee parts, use algorithmic regulations, or classification methods learned on custom features (through, for example, colour clustering or extreme area extraction). MSER (Maximally Stable Extremal Regions) [20] and SWT (Stroke Width Transform) [3] are two candidate proposal methods worth mentioning.

MSER is a blob-definition technique in which pixels are combined into greater blobs using a rolling cutoff. Extremal areas are made by applying a rolling cutoff to cells with intensities greater than the threshold. An extremal region is one in which the pixels in the interior have a pixel with greater or lesser intensity than those on the region's edge. The maximum regions are the ones that maintain a uniform size over numerous thresholds. The smaller maximal regions can then be made offspring of the bigger maximal regions that include them, forming a tree. These trees can be used to identify various items in an image.

The MSER technique is utilised by Chen et al. (2011) to generate several components that can be used as character candidates. Many of the character candidates may be inaccurate because the MSER method was not intended to localise characters in the first place. Because greater candidates might have numerous lesser versions, the overall number of character candidates is substantially enhanced. Many candidates can be eliminated using the computed trees by pruning the tree's child or parent nodes. There are two stages to the pruning process. Initially, vertices in the tree with only one child are trimmed using a sequential cut; just the descendant or parental node survives. This lowering is repetitively done to the whole structure. The fluctuation for various cutoffs of each descendant is then matched to those of other descendant vertices and the parental vertices for each vertex with two or more descendants. Only the detached, specific characters remain after the method tries to return the descendant or parent with the most negligible fluctuation. The single-link clustering technique is used to merge the remaining character candidates into connected components, and feature contenders with the smallest distance between them are put together to form a new feature.

The process is repeated till the length between the final amount exceeds a predetermined limit. The separation between the two components is calculated based on their similarity, which takes into account differences in stroke width and colour as well as physical distance. However, the majority of the related components may not be text and should be eliminated. Extracted features are analysed by a classification model for each of the clusters, which effectively addresses shape and texture features, boundary softness, and the character candidate's length, size and shape, and pixel pitch. By using the classification model, each closely coupled is delegated a possibility of becoming text, and also any aspects with a plausibility less than a predefined cutoff are eliminated.

The SWT is predicated on the idea that texts have a set width of the stroke. It turns a grayscale image into an array of plausible stroke widths for each pixel, which is then sorted into letter candidates if their widths are comparable.

Sliding Window (SW) classification involves rolling windows of different dimensions across an image and doing a binary classification to extract true positive segmented texts. Using morphological procedures [13], Conditional Random Field [2], and other graph-based approaches, these segments will be further categorised into text areas.

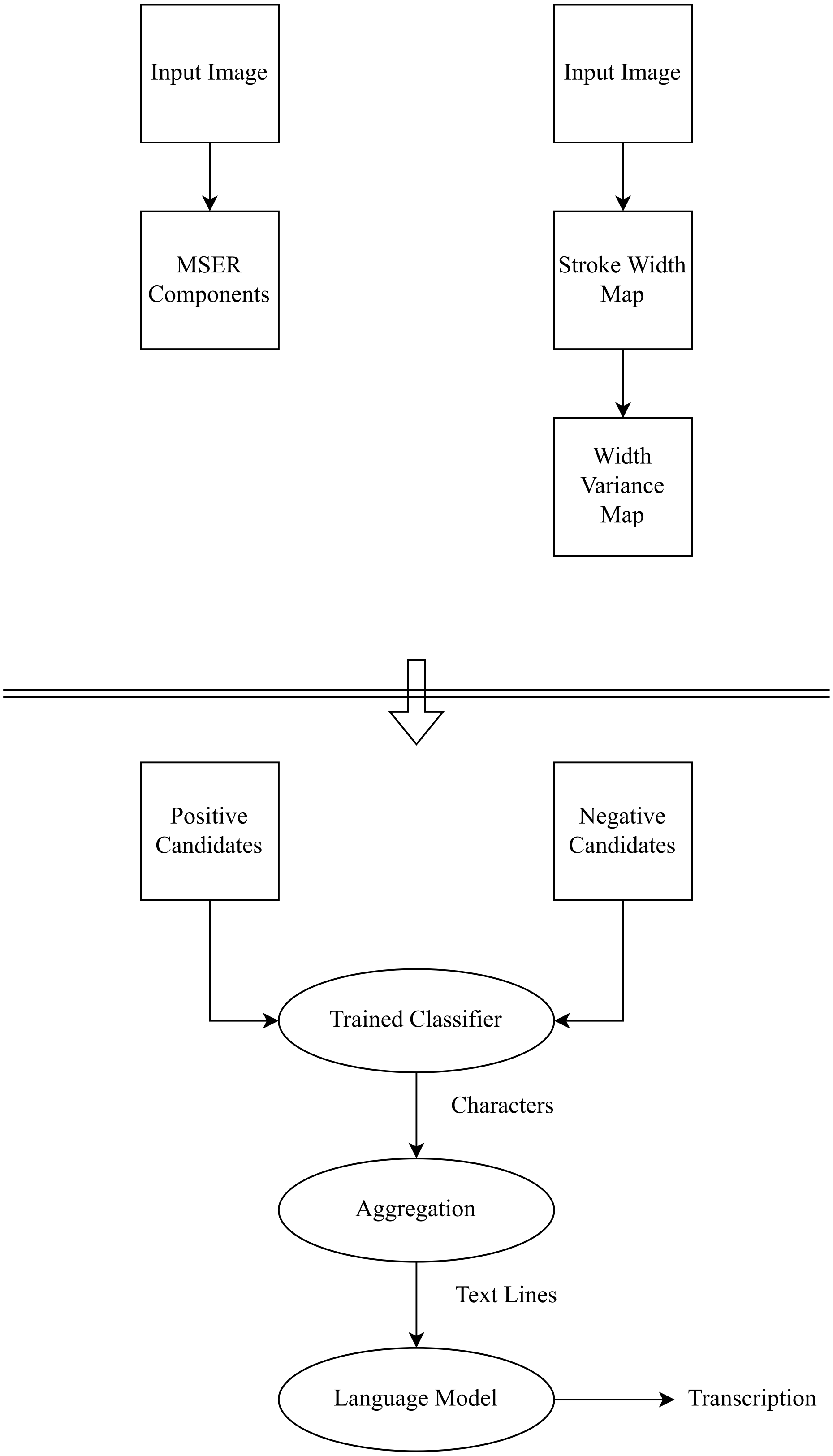


Figure 2.1: Text detection flow before deep learning era.

2.2 Text Detection Methods in the Deep Learning Era

In this section, we will go over some of the most recent developments in deep learning-based text detection. Most approaches in the deep learning era use deep learning-based models, and most academics are attacking the topic from a variety of angles. Deep learning approaches have the advantage of feature learning, which frees us from repeating feature creation for a variety of applications and circumstances. Meanwhile, deep learning's massive representation capacity allows researchers to look at the problem from several angles and approach it in various ways.

Various works have been proposed for detecting text from text images . However, most of these works can be divided into two types of deep learning-based techniques. One falls under the object detection technique, and the other uses the semantic segmentation technique.

2.2.1 Text Detection using Object Detection Technique

Text detection using object detection techniques such as you look only once (YOLO) [21] generally follows the following steps: image is input to a model for training, the model predicts all the bounding boxes and their corresponding classification scores directly, and a loss function is used to penalise the deviation of prediction of the bounding box from that of ground truth, the post-processing step helps to remove all the boxes whose classification score is less than some threshold after the model is trained, and then the remaining redundant bounding boxes are removed by using non-maximum suppression (NMS).

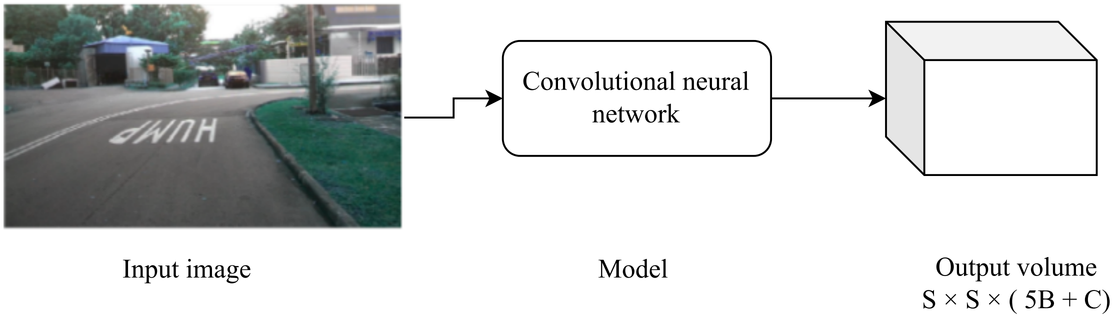
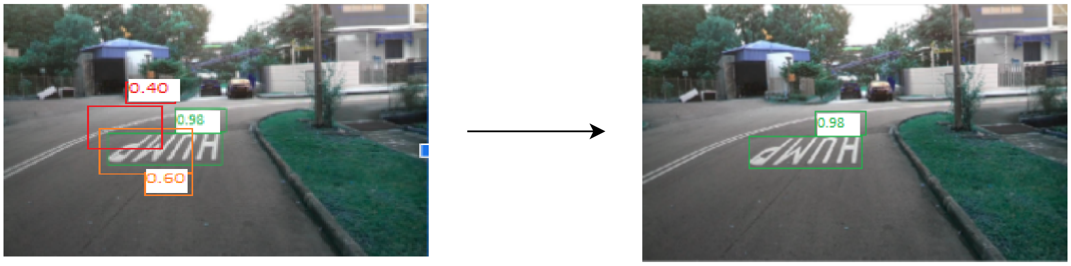


Figure 2.2: Text detection using YOLO [21]

The text detection model training using object detection technique is trained to output S × S × (5B + C) volume. Here, S is the number of grids per row or column. B is the bounding box, and C is the total number of classes. For text detection, the



Before post-processing

After post-processing

Figure 2.3: NMS

number of classes is 2. We require bounding box coordinates as ground truth for the training using the object detection technique.

NMS is the technique used to filter out the proposals. A list of proposal boxes L, matching confidence scores Z, and an overlap threshold T are all inputs to the NMS method. NMS procedure follows the following steps: remove the proposal with the greatest confidence score from L and place it in the final proposal list D (At first, D is empty.), now compare this proposal to all of the others by calculating the *IOU* (Intersection Over Union) of this proposal with all of the others, remove that proposal from L if the *IOU* exceeds the threshold T, remove the proposal with the highest level of confidence from the other proposals in L and add it to D, calculate the *IOU* of this proposition with all of the proposals in L again, and remove the boxes with *IOUs* higher than the threshold, this process is continued until L is devoid of any additional proposals.

Text detection, based upon object detection methods like single shot multibox detector (SSD) [18], YOLO [21], have been employed in TextBoxes [15], which is a word-level detector in the scene image. TextBoxes [15] manipulates the SSD [18] architecture to perform text detection. The main disadvantage of the SSD [18] model in text detection is that it generally predicts a bounding box that does not have a high aspect ratio. Since the texts mostly have a high aspect ratio, Liao et al. [15] designed the different aspect ratios for the default boxes and used 15 convolutional filters, which would better fit the text in the images. Finally, a convolutional recurrent neural network(CRNN) [24] model recognises the detected text from the image. Limitations of TextBoxes [15] include that it could not handle curvy text, arbitrary oriented text and large character spacing between the words. In TextBoxes++ [14], Liao et al. [15] extended TextBoxes [15] to detect multi oriented text instances by introducing oriented ground truth values in their dataset. Shi et al. [23] worked on detecting multi-oriented long text instances by introducing links between the different segmented parts of the single text instance.

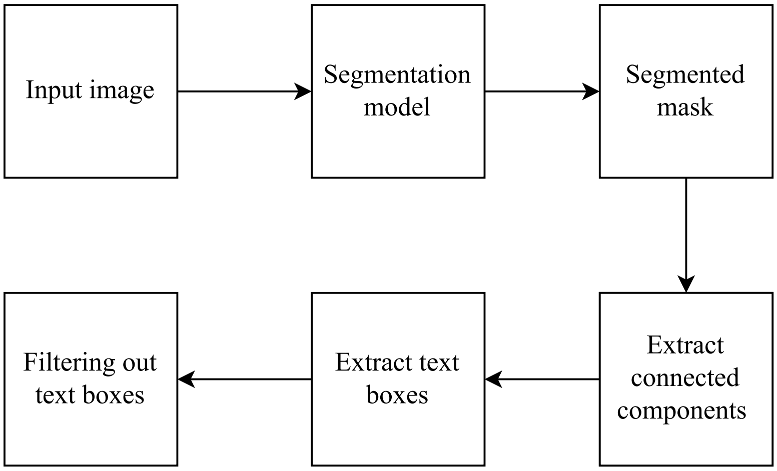


Figure 2.4: Text detection using segmentation technique

2.2.2 Text Detection using Segmentation Technique

Text detection using segmentation generally follows the following steps: image is input to a model for training, the model produces a pixel-level classification score for distinguishing between text and non-text pixels in the image, and a loss function is used to penalise the deviation of the predicted segmented mask from that of the ground truth segmented mask, the post-processing step helps extract the text boxes and then filter out the redundant and unusual aspect ratio boxes after the model is trained. Text detection, based on segmentation methods, has been proposed in EAST [32]. In this paper, pixel-level classification is obtained, and for each corresponding pixel, the distance of pixel position to the four edges of the rectangle along with the rotation angle is predicted. The subsequent step includes generating all the bounding boxes from the output channels, and finally, the NMS process is used to remove all the redundant boxes from the image. For the detection of curved-shaped text, a progressive scale expansion network (PSENet) [27] has been proposed. It deals with two problems: the first problem is that of detecting text using pixel-level classification, it is hard to separate out very close adjacent text regions, and the other problem is due to the limitation of quadrangular representation of the bounding box, it is difficult to fit the curved text perfectly to the text instance. Segmented results are extracted at multiple scales; starting with the smallest segmented result, it uses a progressive scale expansion algorithm to obtain the final shape. This helps in separating text instances quite well. If there are three different segmented results, then we need three different ground truths for the particular scale.

Document text detection is different from scene text detection as text in the document is relatively dense. Because of this, two different words may get detected as a single word and vice versa. DetectGAN [31] was proposed as a character-level detection framework for detecting text from the document images to overcome this problem. The main aim of DetectGAN [31] was to make a conditional generative model (CGAN) [9] using UNET [22] architecture which would generate corresponding directional feature maps for a particular input image and text scores from the camera-captured document image.

A directional feature map is used to tackle the problem of the multidirectional orientation of characters in the image. They created four directional maps, namely, head-up, head-down, head-left, and head-right. During recognition, they can find the directional feature and subsequently can rotate the character for subsequent recognition. The discriminator is used to distinguish between a fake directional feature map and a real feature map. For this, the discriminator has two types of inputs: one input is the image with its corresponding ground-truth directional feature map, and the other is the input image with its generated feature map. The job of the discriminator is to distinguish between fake and real. A simple post-processing technique is used instead of the traditional NMS algorithm, which would help extract the textboxes from the generated bounding boxes inside the directional score map.

The scene text detection method suffers from a class imbalance problem as the text in the scene image is sparse and randomly distributed. Also, the text in the scene image is mostly of varying scale. Due to this, dice loss [10], as used in PSENet [27] is preferred over binary cross-entropy loss [10] for scene text image detection. Dice loss tries to handle the class imbalance problem by maximising the model's precision and recall value. While the text in document images is quite dense, the document text detection method does not suffer from such an imbalance problem. Hence, binary cross-entropy loss [10] is well suited for such a scenario.

While the DetectGAN [31] method is state-of-the-art for detecting text from document images, its robustness can be questioned for detecting text from scene images as the discriminator model uses binary cross-entropy loss [10] to distinguish fake and real features. This can make GAN training quite unstable, so a better regression method should be developed in such a scenario. Even though DetectGAN [31] and PSENet [27] perform well for document text detection and scene text detection individually, no single work addresses text detection in both kinds of images. Therefore, the purpose of this work is to make a text detection scheme that addresses the text detection for both types of images.

2.3 Topics Studied

This section focuses on the crucial deep learning topics to study for performing any deep learning task. We will restrict ourselves to topics relevant to this thesis work.

2.3.1 Convolution

The convolution is used for extracting features from the input data. The convolution operation for extracting the feature map from images can be seen in Fig. 2.6. The kernels slide over the entire image left to right and then top to bottom. At each slide, it calculates a new pixel value by using the following formula:

$$y_{cal} = \sum_{i=1}^{i=k_c} \sum_{j=1}^{j=k_c} I(i,j).K(i,j) \tag{2.1}$$

where I,K are the image patch and kernel of size $k_c \times k_c$.

The convolution operation on a single patch of an image can be seen in Fig. 2.5.

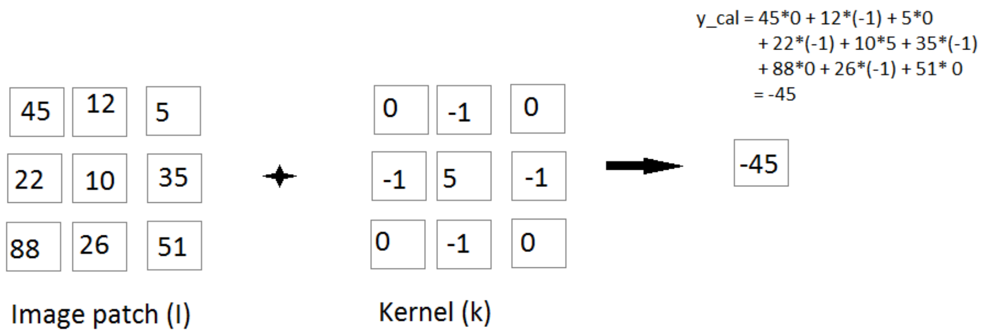


Figure 2.5: Convolution operation on a single patch of the image.

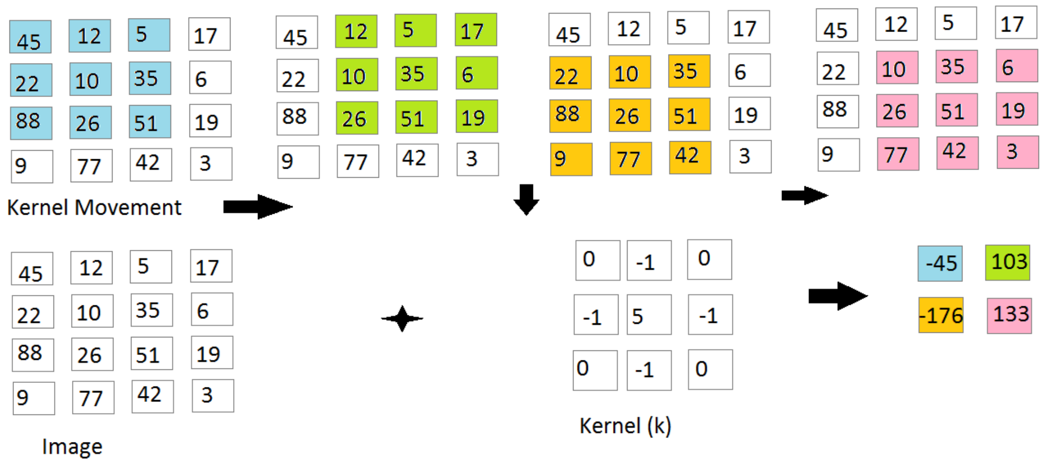


Figure 2.6: Convolution operation on the entire image with a stride of 1 and no padding.

2.3.2 Rectified Linear Unit (RELU)

Single neurons, the basic components of a neural net, jointly calculate a linear combination to locate inputs into outputs. Since a linear function of linear functions continues to remain linear, the system would be unable to portray anything other than a linear correlation between the outputs and inputs. The inclusion of a non-linear component to a system enables this to locate a feature which may match the issue area better than just an essential linear transformation. By its capability to learn network system rapidly, the rectified linear unit (ReLU), $f(k) = maximum(0;k)$, has recently attracted considerable attention. As the functions approach their asymptotes, the gradients of sigmoidal functions tend to be around zero. The ReLU, on the different side, has a slope of one before the component is involved and zero or else, enabling learning to exist regardless of whether the component is involved.

Because the tangent of the ReLU is 1 for all positive attributes, learning will

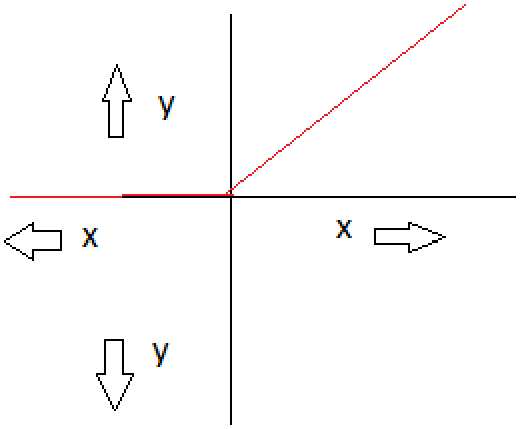


Figure 2.7: RELU

take place in a neuron as long as some feedback results in a positive outcome. This eliminates the requirement to normalise the nonlinearity layer's input, which is required for some sigmoidal functions.

2.3.3 Max-pooling

Max-pooling layer helps to lower the dimensionality of the features obtained by choosing only the highest intensity pixel from a local group of a pixel. The max-pooling operation is illustrated in Fig. 2.8. Here the local group, i.e. window size, is 2×2. In the Max-pooling operation, the window size decides the amount of downscaling to be done. Here, in Fig. 2.8, the window size is 2; hence the features are lowered by a multiple of 2.

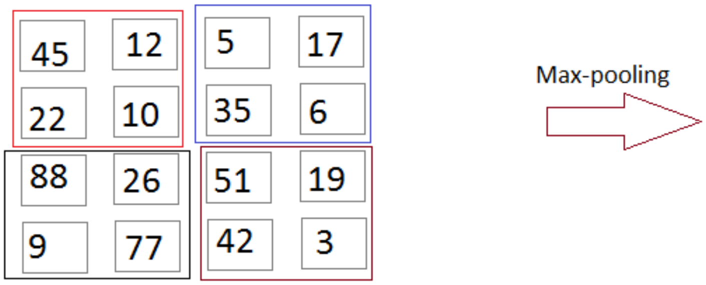


Figure 2.8: Max-pooling operation with the window size of 2 and a stride of 2.

2.3.4 Upsampling using Nearest Neighbour Interpolation

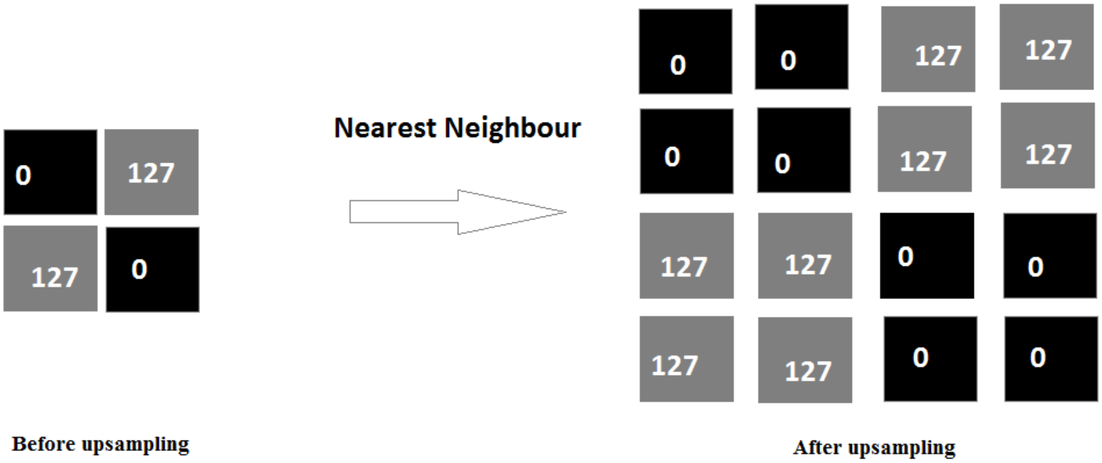


Figure 2.9: Upsampling operation using nearest neighbour interpolation

Upsampling helps increase the dimensionality of the features obtained from the previous layer. It is also useful for reconstructing the downsampled feature maps to visualize the features chosen by the deep learning models for giving its prediction. There are various Upsampling techniques available, but to simply understand how upsampling works, let us take an example of Upsampling using Nearest neighbour interpolation. As shown in Fig. 2.9, the upsampling factor here is 2. The 22 grid in Fig. 2.9 is upsampled to 44 grid in Fig. 2.9. In nearest neighbour interpolation, the nearest neighbour pixel simply gets copied. Hence, this method does not involve a complex calculation of interpolating the pixel values. The other interpolation technique that can be used is bilinear interpolation and bicubic spline interpolation.

2.3.5 Batch Normalization

Batch normalisation attempts to balance the network activations between layers in batches, resulting in batches with a mean of 0 and a variance of 1. Normally, batch normalisation is written as follows:

$$y = \frac{x - E(x)}{\sqrt{var(x) + \epsilon}} \times \gamma + \beta \tag{2.2}$$

The mean and variance of the input vector x are represented by $E(x)$ and $var(x)$, where B_s is the size of x. γ, β are learnable parameter vectors of size B_s. For each batch, as well as each dimension/channel, the mean and standard deviation are computed. γ, β can be used to scale and move the normalised value, allowing us to modify the shape of the data as we progress through the layers.

Need Of Batch Normalization

Batch Normalization is generally used because of following reasons:

1. Reduce unnecessary covariate shift, which is defined as "a change in the pattern of network activations as a result of changes in network parameters during training.".

2. Training will be completed more quickly. Batch normalisation decreases intrinsic covariate shift and fixes the pattern of network activations, allowing the network to learn at higher rates and train faster.

3. The network will be more regular as a result of this.

2.4 Chapter Summary

In this chapter, we looked at various techniques of text detection before and during the deep learning era. Prior to the deep learning era, text detection primarily focused on extracting low-level or mid-level hand-crafted characteristics. To obtain good performance, they require intricate and repetitive pre-and post-processing, and they are sensitive to a variety of environments. In the deep learning era, text detection is done using segmentation based and object detection based techniques. Pixel level classification is done for text detection using a segmentation-based method. For the object detection based technique, the bounding box is regressed to find the locality of the text instance. After discussing both kinds of techniques, various works related to both kinds of techniques are briefly reviewed. The main advantage of using these techniques is the learning of features and is less sensitive to a variety of environments as compared to traditional methods. Then, we looked in-depth at various deep learning topics that are required for subsequent upcoming chapters. The next chapter focuses on studying the PSENet model for text detection.

CHAPTER 3
Progressive Scale Expansion Network

We will look at the PSENet model in this chapter. We selected PSENet for the work of text detection in images because of their recent rise to prominence. The next chapter shows the proposed SDINet scheme applied to the PSENet model for text detection in scene images and document images.

Text detection has developed significantly with the current advancements of CNN models. However, the algorithm has two obstacles that prohibit it from being used in industry. Firstly, most methods rely on an inexact quadrilateral bounding box to locate texts of any shape. On the other side, two text objects that are near to one other may result in a false detection that encompasses both instances. They introduced the PSENet model to address these two issues by accurately detecting text instances with various forms. PSENet generates various kernel scales for every text object, progressively expanding the smallest scale result to the text instance with the whole shape.

3.1 Ground Truth Generation for PSENet Model

PSENet predicts segmentation results (e.g. S_1, S_2,..., S_n) with various kernel scales. As a result, it is necessary to have the relevant known segmented data with varied kernel scales during training. By decreasing the true text instance, these known data labelling can be done quickly and effectively.

The true text object is represented by the polygon with the biggest segmentation mask in Fig. 3.1. The Vatti clipping technique is used to reduce the true polygon p_n by d_i pixels and get smaller polygon p_i in Fig. 3.1. Following that, for segmentation label ground truths, each reduced polygon p_i is converted into a 0/1 mask. These known truth maps are referred to as G_1,G_2,..., G_n. The margin d_i between p_n and p_i may be computed if we regard the scale ratio as r_i.

$$d_i = \frac{Area(p_n) \times (1 - r_i^2)}{Perimeter(p_n)} \tag{3.1}$$

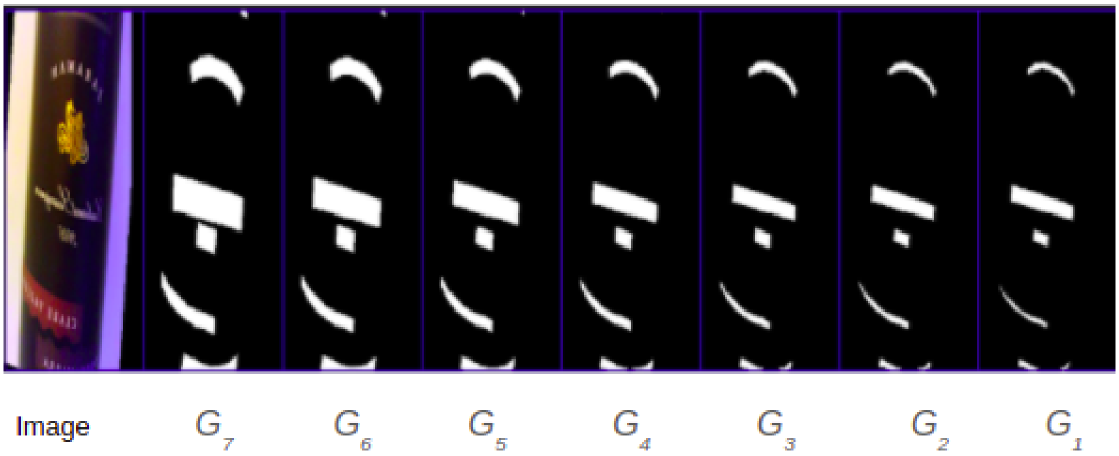


Figure 3.1: PSENet ground truths visualization

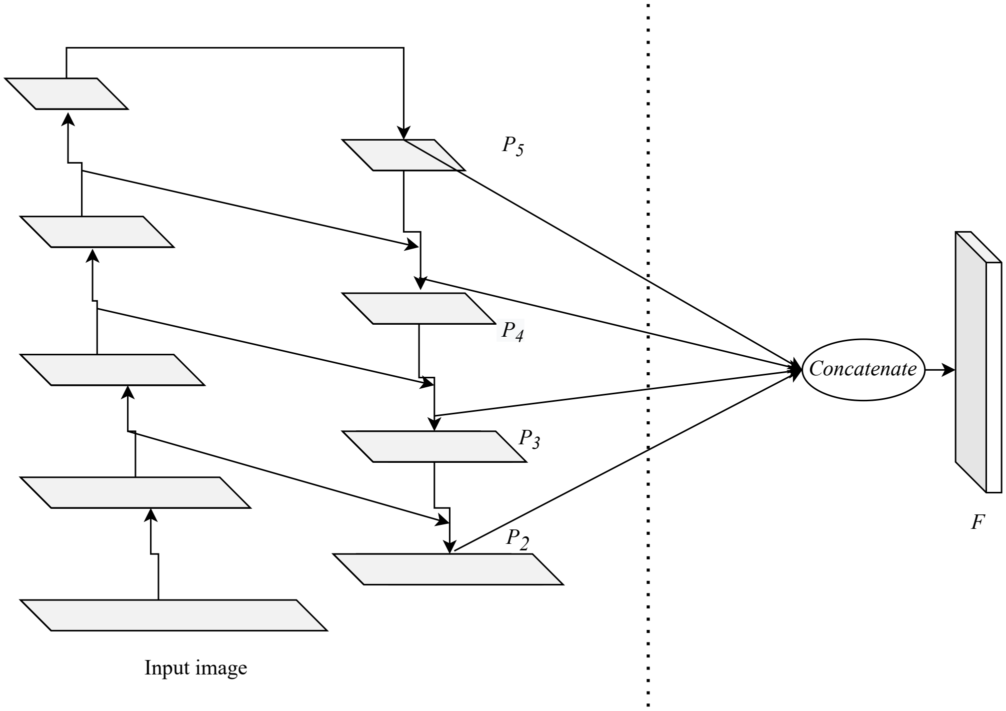


Figure 3.2: Progressive scale expansion model

Where $Perimeter()$ is the method for calculating the polygon circumference, and $Area()$ is the method for calculating the amount of region covered by the polygon. We also define the scale ratio r_i for the known data map G_i as follows:

$$r_i = 1 - \frac{(1-a) \times (n-i)}{n-1} \tag{3.2}$$

where a is the smallest scale ratio, which is a number between 0 and 1. According to Eqn. 3.1, the scale ratio values (i.e., r_1, r_2,..., r_n) are determined by two hyperparameters, n and a, and they rise sequentially from a to 1.

3.2 PSENet Model

PSENet's fundamental framework is based on FPN [16]. The backbone provides it with four 256-channel features (P_2, P_3, P_4, P_5). To incorporate the feature representations from bottom to top levels, it merges the four feature maps to just get convolutional feature F with 1024 channels as shown below:

$$F = P_2 || Up_2(P_3) || Up_4(P_4) || Up_8(P_5) \tag{3.3}$$

where "||" denotes concatenation and $Up_2()$, $Up_4()$, and $Up_8()$ denote 2, 4, and 8 times upsampling, respectively. After that, F is given as input to Conv(3, 3)-BN-RELU layers, and the number of channels is lowered to 256. It then goes through n Conv (1, 1)-Up-Sigmoid layers produce S_1, S_2,..., S_n segmentation results. Convolution [12], rectified linear units [4], batch normalisation [7] and upsampling are the terms used for Conv, RELU, BN and Up, respectively.

The different segmentation results are converted to a single segmentation result by using a progressive scale expansion which is dependent on the BFS technique. Let's say we have four segmentation results $S = \{S_1, S_2, S_3, S_4\}$ and three connected components for each segmentation prediction, then for each connected components starting with the smallest segmentation kernel, it iteratively merges the adjacent pixels. It's important to note that certain pixels may become incompatible during the expansion process. According to the concept, the clashing pixel can only be fused by a single kernel on a first-come, first-served basis.

3.3 Chapter Summary

In this chapter, the architecture and working of the PSENet model are studied in depth. We first looked at how the ground truth is generated by shrinking the polygons at various scales. After that, we looked at the PSENet model making multiple segmentation predictions for its respective multiple ground truths. Then we looked at the different layers and activation functions used in the PSENet model. We also looked at the process of converting multiple predictions into a single prediction. In the following chapter, we show and discuss our proposed method using the PSENet model for our task.

CHAPTER 4

Proposed Text Detection Scheme

Before our proposed text detection scheme, an approach was tried where we first tried to impose an attention mechanism for detecting text from images. The idea was to attain the scene text and document text features, but that did not work well because it became difficult to know when to impose such attention. Then our approach was tried where rather than finding where to impose such attention, we separately designed a classification model that will extract scene and document features and make the decision about whether the input image is scene text type or document text type. Then the model is penalized according to the input image type by using a weighted loss. The idea behind this is that such a loss function will better penalize the model by covering scene text and as well as document text loss.

Our Proposed SDINet scheme can be seen in Fig. 4.1. During training, inspired by PSENet [27], our SDINet tries to predict n segmentation results at multiple scales. Apart from this, our scheme uses the classification model that is designed to classify input image as scene text or document text. S_1, S_2, ... , S_n are the predicted segmentation results each increasing in scale from S_1 to S_n. G_1, G_2, ... , G_n are the ground truth segmentations each increasing in scale from G_1 to G_n. In our scheme, weighted loss as discussed in Section 4.1 has been used for better updating the training parameters during backpropagation. The classification model is trained for this purpose separately for classifying an input image as a scene text type image or document text type image. The classification model in Fig. 4.2 helps in the entire training process by providing important weight factors O_s and O_d. The last two neurons in Fig. 4.2 represents O_s and O_s, respectively. O_s is the probability that the input image is a scene text type image, O_d is the probability that the input image is a document text type image.

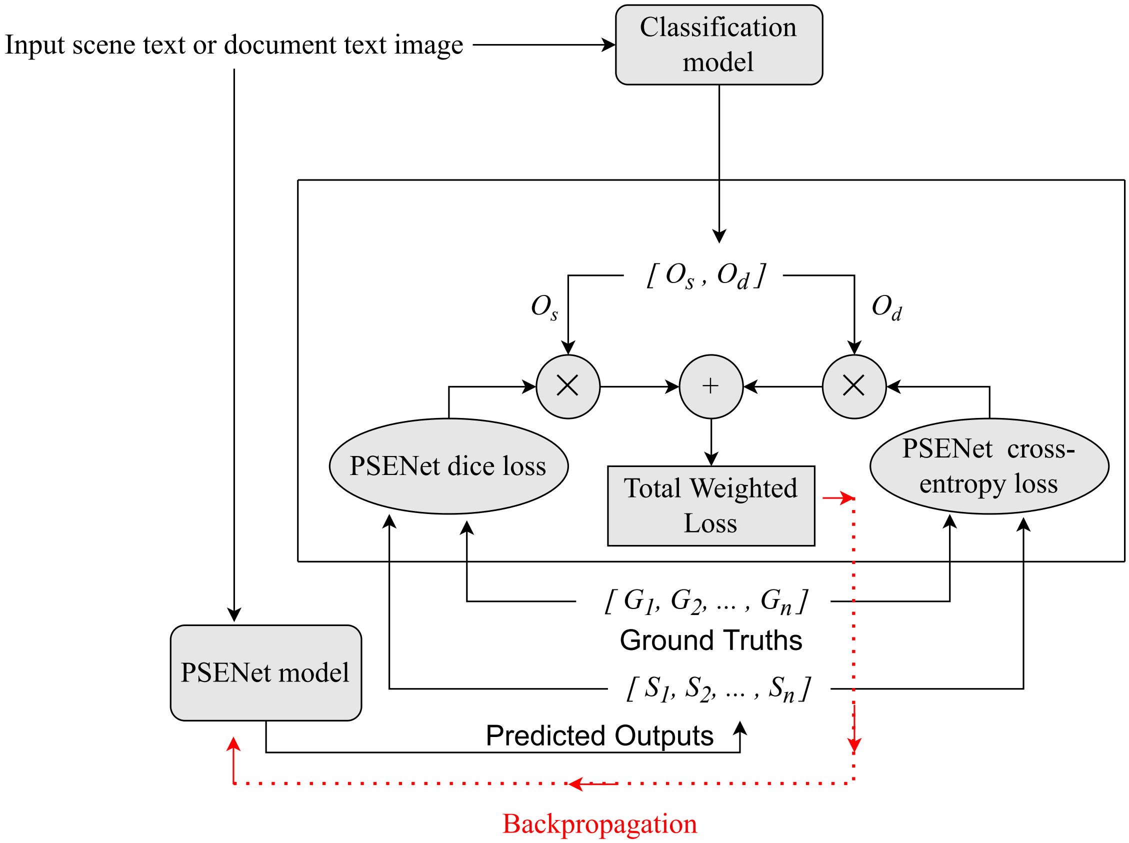


Figure 4.1: Our SDINet scheme

4.1 Weighted Loss

During training, let the dimension of all the segmentation masks be $M \times N$. The constraints for the weight factors O_s and O_d can be seen below:

$$\begin{aligned} 0 \leq O_s \leq 1 \\ 0 \leq O_d \leq 1 \\ O_s + O_d = 1 \end{aligned} \tag{4.1}$$

The weighted loss designed is shown below:

$$L = O_s L_{ol} + O_d L_{bl} \tag{4.2}$$

L_{ol} and L_{bl} are the PSENet dice loss and PSENet cross-entropy loss, respectively. L_{ol} and L_{bl} are calculated using following equations:

$$L_{ol} = \alpha L_{ctd} + (1 - \alpha) L_{std} \tag{4.3}$$

$$L_{bl} = \beta L_{ctb} + (1 - \beta) L_{stb} \tag{4.4}$$

L_{ctd}, L_{std}, L_{ctb} and L_{stb} are the complete text instance dice loss, shrunk text instance dice loss, complete text instance binary cross-entropy loss and shrunk text instance binary cross-entropy loss, respectively. α and β are the tunable hyperparameter. L_{ctd}, L_{std}, L_{ctb} and L_{stb} are calculated using below equations:

$$L_{ctd} = 1 - D(S_n, G_n) \tag{4.5}$$

$$L_{std} = 1 - \frac{\sum_{i=1}^{n-1} D(S_i, G_i)}{n-1} \tag{4.6}$$

$$L_{ctb} = B(S_n, G_n) \tag{4.7}$$

$$L_{stb} = \frac{\sum_{i=1}^{n-1} B(S_i, G_i)}{n-1} \tag{4.8}$$

$D(S_n, G_n)$ in Eqn. (4.5) is the dice coefficient function between predicted complete text instance and ground truth complete text instance segmentation mask. As shown in Eqn. (4.6), $D(S_i, G_i)$ is the general dice coefficient function between S_i and G_i. S_i is the i^{th} predicted segmentation result and G_i is the i^{th} ground truth segmentation. $D(S_i, G_i)$ is calculated using following equation:

$$D(S_i, G_i) = \frac{2 \sum_{x,y} (S_{i,x,y} * G_{i,x,y})}{\sum_{x,y} S_{i,x,y}^2 + \sum_{x,y} G_{i,x,y}^2} \tag{4.9}$$

Here, $S_{i,x,y}$ and $G_{i,x,y}$ denotes the value of the i^{th} segmentation mask at the (x, y) position for the predicted segmentation mask and ground truth segmentation mask, respectively. $B(S_n, G_n)$ in Eqn. (4.7) is the binary cross entropy loss function between predicted complete text instance and ground truth complete text instance segmentation mask. $B(S_i, G_i)$ in Eqn. (4.8) is the general binary cross entropy loss function between i^{th} predicted and ground truth segmentation text instance. $B(S_i, G_i)$ is calculated using below equations:

$$B(S_i, G_i) = -\frac{\sum_{x=1}^{M} \sum_{y=1}^{N} (B_1 + B_2)}{MN} \tag{4.10}$$

where,

$$B_1(G_{i,x,y}, S_{i,x,y}) = G_{i,x,y} log(S_{i,x,y}) \tag{4.11}$$

$$B_2(G_{i,x,y}, S_{i,x,y}) = (1 - G_{i,x,y}) log(1 - S_{i,x,y}) \tag{4.12}$$

4.2 Classification Model

The weight factors O_s and O_d are calculated using a classification model, see Fig. 4.2. The classification model is trained to classify an input image as a scene text type image or document text type image. In our classification model, two convolution layers are used. Each convolution layer is followed by the RELU function and max-pooling layer. A convolution operation helps us to extract important features that are then supplied to the next layer. RELU helps to activate specific features extracted from the convolution layer. The max-pooling layer helps to reduce the dimensionality of data obtained from the previous layer. After extracting features from the image, a dense connection is applied. A dense connection in a neural network is where all the neurons from the previous layer are connected to every neuron in the current layer. The final connected layer produces C_{co} =$\{O_1, O_2\}$. In order to fulfil the criteria mentioned in Eqn. (4.1), C_{co} is passed through a softmax function that gives us final output as C_{cf}=$\{O_s, O_d\}$. During the training of PSENet [27], each input image is also given as input to this classification model. The elements of output C_{cf} are accordingly substituted in Eqn. (4.2).

Loss of classification Model

The cross-entropy loss is used as loss function for our classification model. Let $G^i_{cf} = \{G_s, G_d\}$ and $C^i_{cf} = \{O_s, O_d\}$ be the i^{th} ground truth and predicted training sample, respectively. Here, $G_s, G_d \in \{0, 1\}$. K is the total number of the training sample. The loss function L_c obtained can be seen in Eqn. (4.15).

$$L^i_{c1}(G^i_{cf}, C^i_{cf}) = G_s log(O_s) \tag{4.13}$$

$$L^i_{c2}(G^i_{cf}, C^i_{cf}) = G_d log(O_d) \tag{4.14}$$

$$L_c = -\frac{\sum_{i=1}^{K}(L^i_{c1} + L^i_{c2})}{K} \tag{4.15}$$

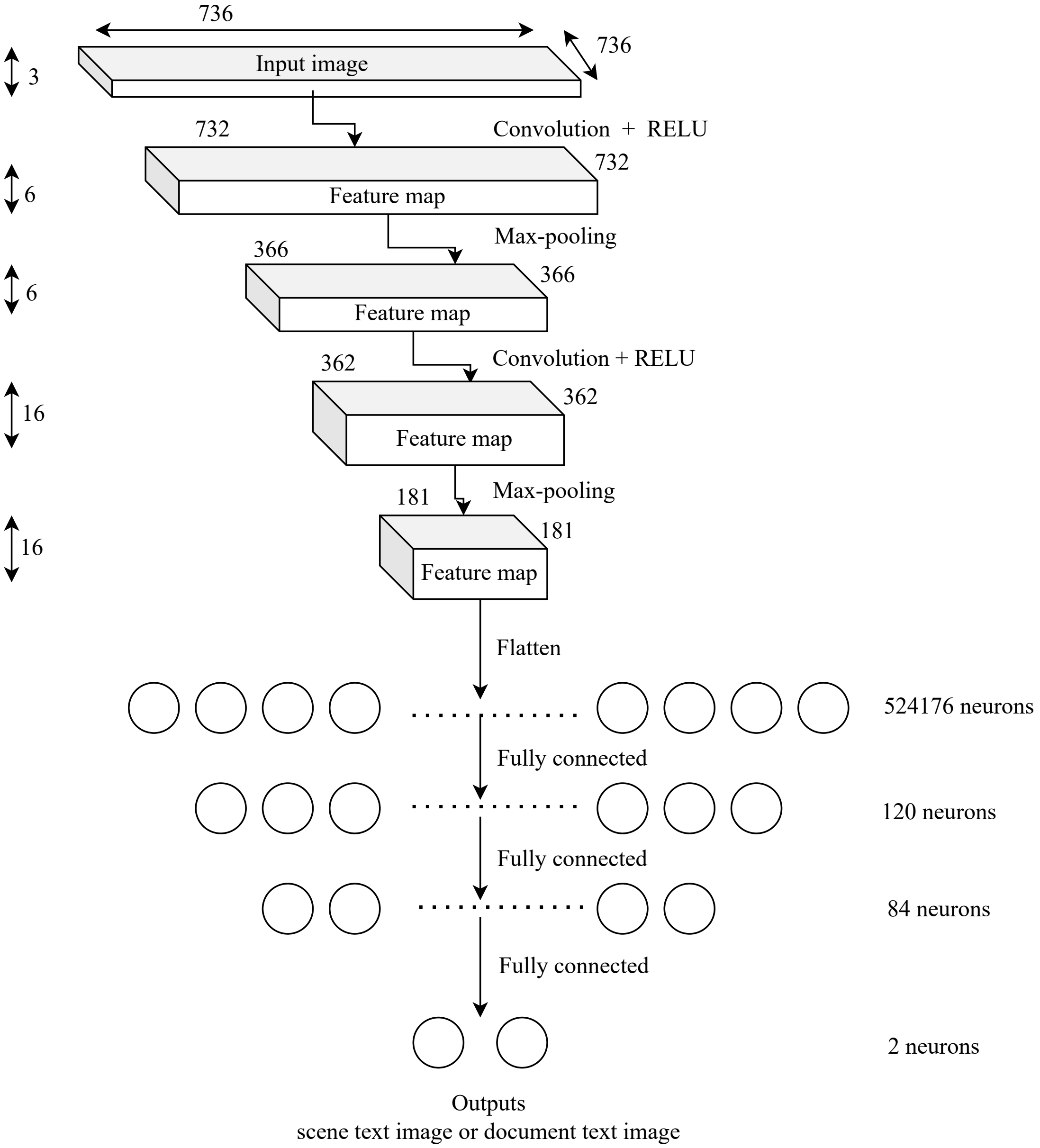


Figure 4.2: Classification model that classifies an input image as a scene text type image or document text type image.

4.3 Algorithm of the Proposed SDINet Scheme

The procedure of the proposed SDINet scheme on the PSENet model is shown below:

Algorithm 1 Algorithm of the SDINet scheme

```
Require: Input image $I$, Ground truths $G = \{G_1, G_2, ..., G_n\}$, total epochs $E$. $M(.)$
    and $M_c(.)$ are the functions for calculating the output of PSENet and classification model.
Ensure: Segmentation results $S = \{S_1, S_2, ..., S_n\}$
    $E_p \leftarrow E$
    $e \leftarrow 0$

        1. Start
        2. Randomly initialize all the training parameters.
        3. Repeat steps 4 to 11 until $e \leq E_p$ else, go to step 12.
        4. $e \leftarrow e + 1$
        5. $O_s, O_d \leftarrow M_c(I)$
        6. $O \leftarrow \{O_s, O_d\}$
        7. $S_1, S_2, ..., S_n \leftarrow M(I)$
        8. $S \leftarrow \{S_1, S_2, ..., S_n\}$
        9. Calculate the total loss $L$ using $O$, $S$, and $G$ by following Eqn. 4.2
       10. Update the training parameters of the PSENet model by minimizing $L$.
       11. Go to step 3
       12. End
```

The multiple predicted segmentation results S are converted to a single segmentation result, as discussed earlier. In the end, the minimum rectangle needed to extract the contours from the image is estimated as our text bounding box. All those contours whose area is smaller than some threshold are appropriately filtered out to decrease the false detection of text in the image.

4.4 Chapter Summary

In this chapter, we looked at our proposed text detection scheme using the PSENet model. We first looked at the entire text detection training scheme proposed that helps to better update the training parameter according to the type of input image. Then, the classification model that helps us to find the important weight factors by classifying an input image as scene text or document text type is explained. In the end, the algorithm of the proposed SDINet scheme on the PSENet model is presented. The next chapter focuses on the implementation results and comparison with various text detection methods.

CHAPTER 5

Implementation Results and Comparisons

For evaluating the performance of the proposed scheme, a combination of scene text and document text datasets is used because, to the best of our knowledge, there is no standard dataset available that contains a combination of both types of text images. SROIE 2019 [6] and SCUT-CTW 1500 [30] datasets are combined to generate a hybrid dataset. SCUT-CTW 1500 [30] is a scene text image dataset that also contains curved text. SCUT-CTW 1500 [30] has a total of 1500 text images. Out of which 1000 are training images, and 500 are test images. SROIE 2019 [6] is a dataset from ICDAR 2019 challenge for recognising and detecting text from scanned documents. This dataset is challenging as there are a lot of folded pages, poor ink quality, distorted images etc. SROIE 2019 [6] has a total of 1000 images, out of which the number of training images are 640, and the number of test images are 360. For the classification model, the scene text images are labelled as 1, and document text images are labelled as 0 and are trained independently of the PSENet [27] model. For the PSENet [27] model, the text annotation is provided in both datasets. For training the models, the experimental development environment is as follows: CPU: Intel(R) Xeon(R) 2.30GHz, RAM: 12GB, GPU: Tesla P100-PCIE 12GB, and deep learning framework: PyTorch.

The images in the hybrid dataset are resized to 736×736, and the PSENet [27] model is trained using our SDINet scheme up to 500 epochs. The batch size is set to 4. Initially, we set the learning rate to 0.001, and it is lowered to 0.0001,0.00001 after the 200^{th} epoch and 400^{th} epoch, respectively. α, β are set to 0.7 and 0.7, respectively. The smallest kernel size is set to 0.7, and 6 different kernels are used in the whole process. The classification model is trained up to 20 epochs with a learning rate of 0.001 and batch size of 4.

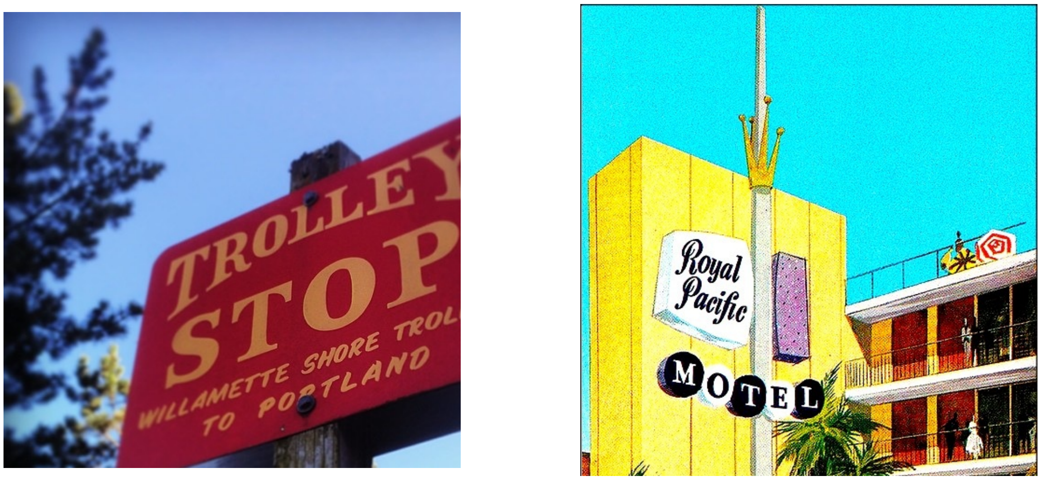
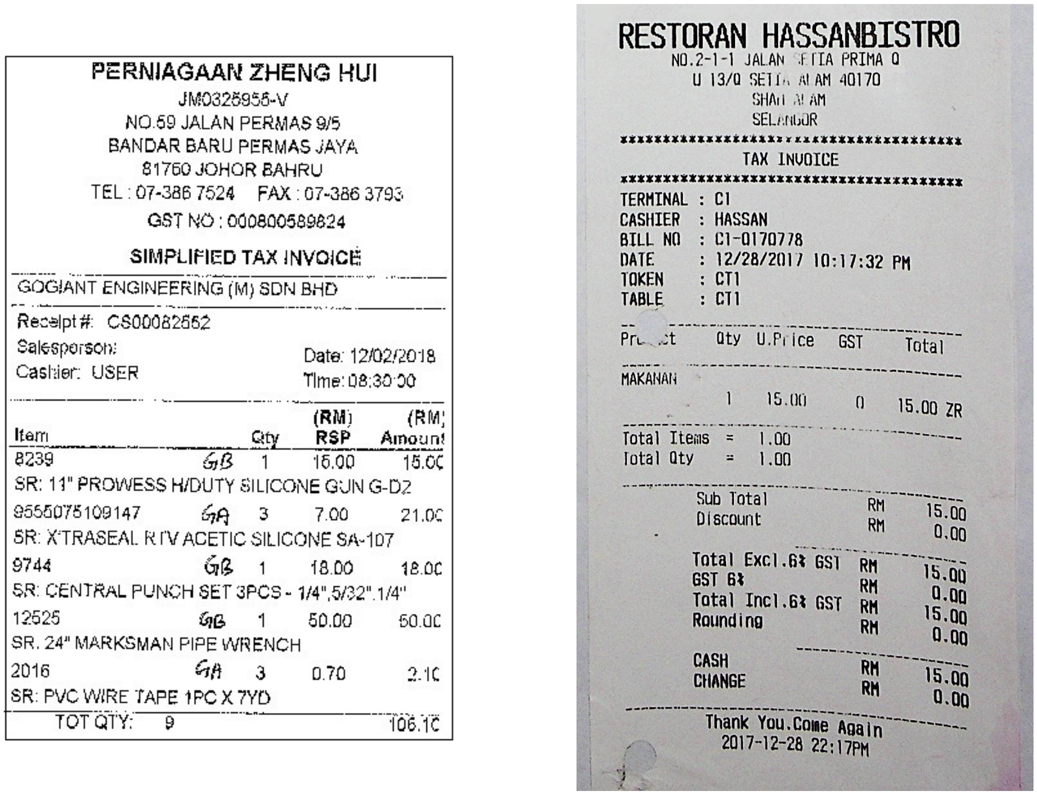


Figure 5.1: Example images from the SCUT-CTW 1500 [30] dataset.



```
PERNIAGAAN ZHENG HUI
JM0325955-V
NO.59 JALAN PERMAS 9/5
BANDAR BARU PERMAS JAYA
81760 JOHOR BAHRU
TEL : 07-386 7524     FAX : 07-386 3793
GST NO : 000800589824

SIMPLIFIED TAX INVOICE
GOGIANT ENGINEERING (M) SDN BHD
Receipt #:  CS00082552
Salesperson:
Cashier:  USER                    Date: 12/02/2018
                                  Time: 08:30:00
                                (RM)      (RM)
Item                   Qty      RSP       Amount
8239             GB     1      15.00      15.00
SR: 11" PROWESS H/DUTY SILICONE GUN G-D2
9555075109147    GA     3       7.00      21.00
SR: X'TRASEAL RTV ACETIC SILICONE SA-107
9744             GB     1      18.00      18.00
SR: CENTRAL PUNCH SET 3PCS - 1/4",5/32",1/4"
12525            GB     1      50.00      50.00
SR. 24" MARKSMAN PIPE WRENCH
2016             GA     3       0.70       2.10
SR: PVC WIRE TAPE 1PC X 7YD
     TOT QTY:    9                        106.10
```

```
RESTORAN HASSANBISTRO
NO.2-1-1 JALAN SETIA PRIMA Q
U 13/Q SETIA ALAM 40170
SHAH ALAM
SELANGOR
****************************************
TAX INVOICE
****************************************
TERMINAL  : C1
CASHIER   : HASSAN
BILL NO   : C1-0170778
DATE      : 12/28/2017 10:17:32 PM
TOKEN     : CT1
TABLE     : CT1
----------------------------------------
Product    Qty  U.Price   GST     Total
----------------------------------------
MAKANAN
            1    15.00     0     15.00 ZR
----------------------------------------
Total Items  =   1.00
Total Qty    =   1.00
----------------------------------------
        Sub Total              RM   15.00
        Discount               RM    0.00
                   ---------------------
        Total Excl.6% GST      RM   15.00
        GST 6%                 RM    0.00
        Total Incl.6% GST      RM   15.00
        Rounding               RM    0.00
                   ---------------------
        CASH                   RM   15.00
        CHANGE                 RM    0.00
----------------------------------------
        Thank You.Come Again
        2017-12-28 22:17PM
```

Figure 5.2: Example images from the SROIE 2019 [6] dataset.

5.1 Evaluation Protocol

Evaluation of the model is done after training the model. Evaluation of the model is necessary because it helps us to know the performance of the model. The model is evaluated by using precision Eqn. 5.1, recall Eqn.5.2 and F-score Eqn. 5.3 values. Let precision, F-score, and recall be P, F, and R.

$$P = X_p/(X_p + Y_p) \tag{5.1}$$

$$R = X_p/(X_p + Y_n) \tag{5.2}$$

$$F = 2PR/(P + R) \tag{5.3}$$

Y_n, X_p, Y_p represents false negative, true positive, and false positive, respectively. IOU is used for matching the known bounding box and predicted box. IOU measures the amount of overlap between the known bounding box and the predicted bounding box. More formally, for a known rectangle A and predicted rectangle B, IOU is defined as follows:

$$IOU = A \cap B/(A \cup B) \tag{5.4}$$

5.2 Performance Comparison between the SDINet scheme and PSENet

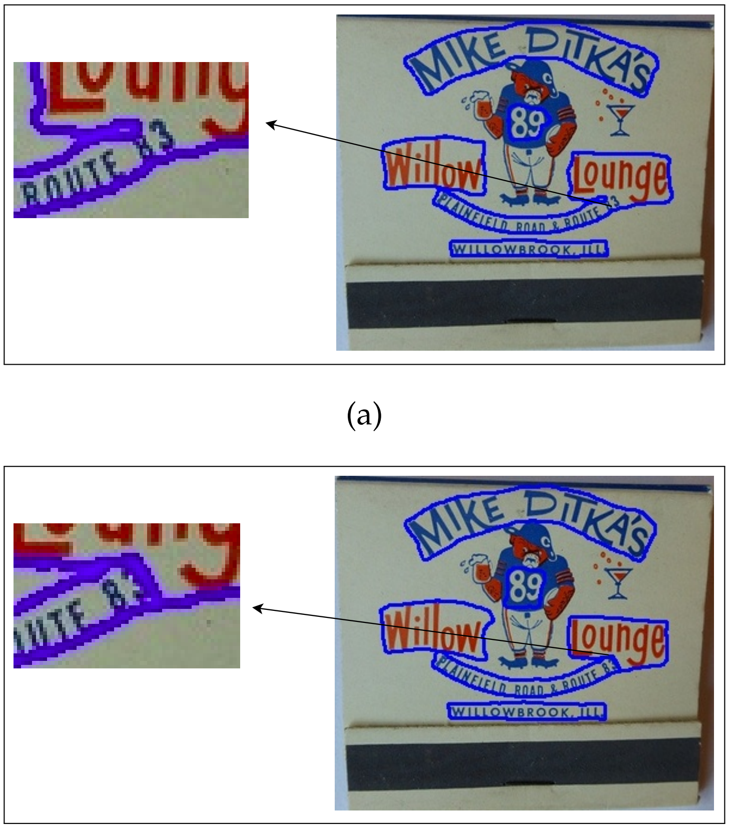
For comparison of SDINet scheme with the PSENet [27], the original PSENet [27] is trained with the same training configuration as discussed above, and everything is trained from scratch. The loss and the IOU curve of SDINet scheme over epochs during training can be seen in Fig. 5.4. The loss curve is decreasing, and the IOU curve is increasing over time which suggests that the training is stable. After the training, models are tested on test images of SROIE 2019 [6] and SCUT-CTW 1500 [30] dataset individually.

Method	Precision (%)	Recall (%)	Fscore (%)
PSENet original	82.39	79.30	80.82
SDINet scheme (ours)	82.95	80.54	81.73

Table 5.1: Performance comparison for SCUT-CTW 1500 [30] dataset when trained on both types of input images.

Method	Precision (%)	Recall (%)	Fscore (%)
PSENet original	97.30	94.91	96.01
SDINet scheme (ours)	97.50	95.00	96.23

Table 5.2: Performance comparison for SROIE 2019 [6] dataset when trained on both types of input images.



(a)

(b)

Figure 5.3: Detection results when trained on both types of input images for scene text type images: (a) Detection by PSENet; (b) Detection using SDINet scheme.

The proposed SDINet scheme applied on the PSENet model performs better in all the parameters of the evaluation for both the scene text and document text dataset. Specifically, the recall is increased by more than 1%, and F-score is increased by approximately 1% for SCUT-CTW 1500 dataset [30], respectively. The full comparison result for SCUT-CTW 1500 [30] and SROIE 2019 [6] dataset can be seen in Tables 5.1 and 5.2.

The performance comparison on scene text type images can be seen in Fig. 5.3. From Fig. 5.3(a), it can be seen that when trained on both types of input images, the PSENet [27] loses its performance for separating the text instances that are very close to each other. The proper separation of detected text instances low-

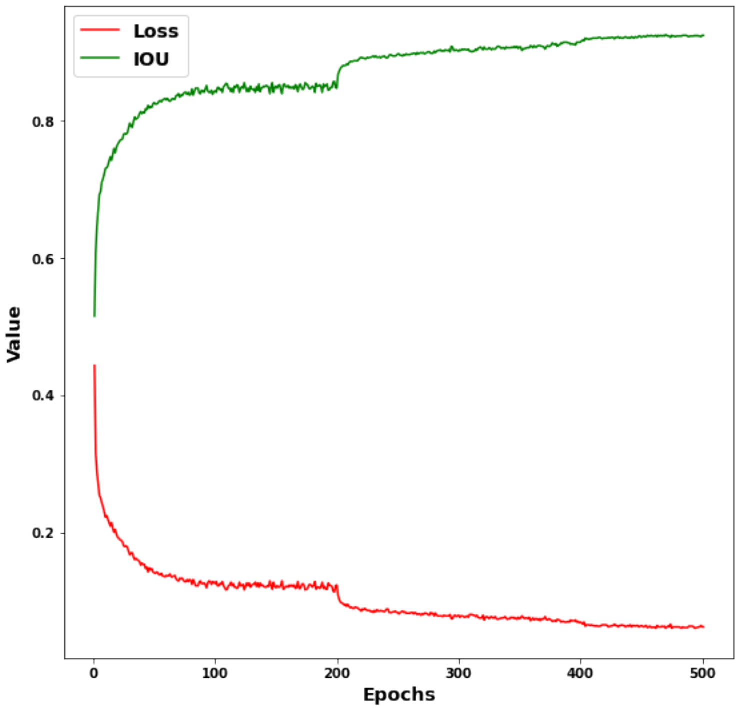
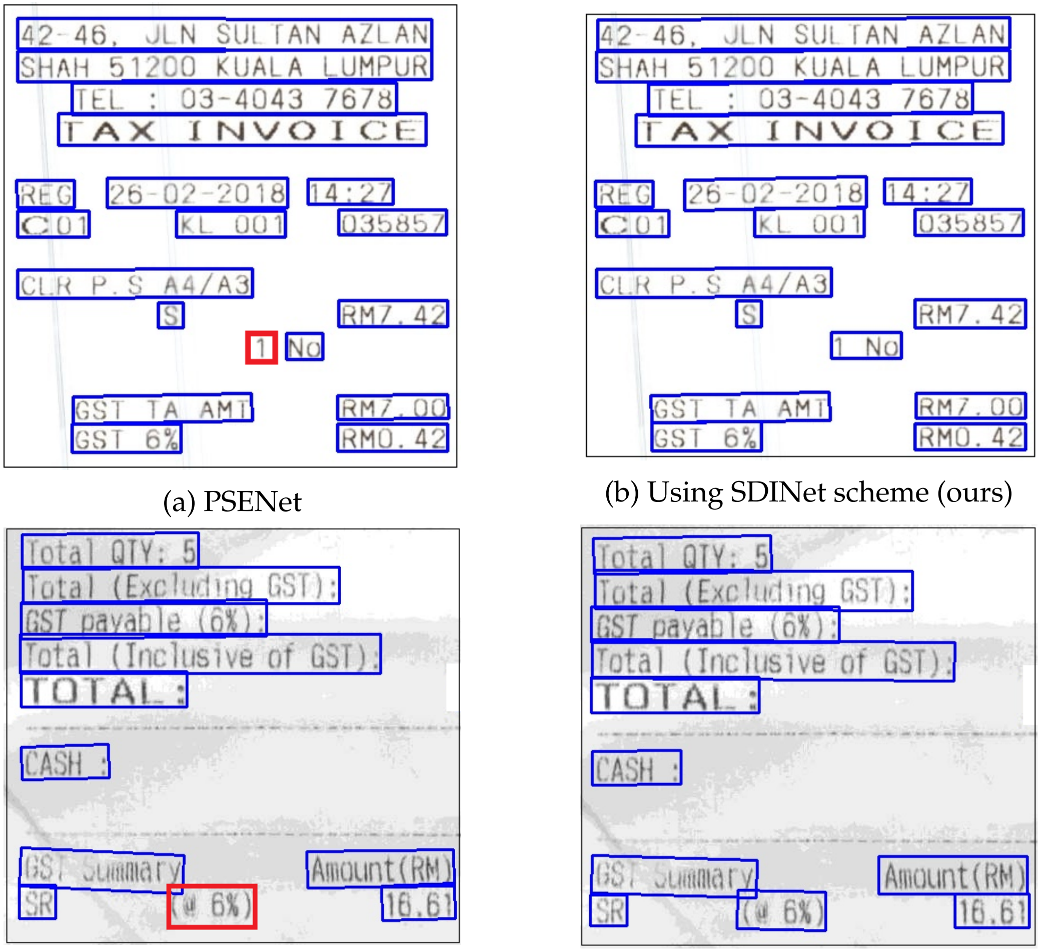


Figure 5.4: *IOU* and Loss curve for the SDINet scheme

ers the missed detection of text in an image that, in turn, aids in the subsequent text recognition process. Meanwhile, thanks to the proposed SDINet scheme, the PSENet [27] model regains the performance for separating the text instance that is very close to each other, see Fig. 5.3(b). The performance comparison on document text type images can be seen in Fig. 5.5. The detection results obtained by the PSENet model are shown in Fig. 5.5(a) and 5.5(c), respectively. Fig. 5.5(b) and 5.5(d) shows the detection result obtained by SDINet scheme. It can be clearly seen that the proposed scheme performs better compared to PSENet as it has lesser missed detection.



(a) PSENet

(b) Using SDINet scheme (ours)

(c) PSENet

(d) Using SDINet scheme (ours)

Figure 5.5: Detection results when trained on both types of input images for document text type images. Missed detection of PSENet is shown in the red text box.

5.3 Comparison with other Text Detection Algorithms

For the comparison of the proposed scheme with other text detection algorithms, the models is trained individually on SROIE 2019 [6] and SCUT-CTW 1500 [30] dataset, respectively. The performance comparison when trained on only individual type of input images is shown in Table 5.3 and Table 5.4. From Table 5.3 and Table 5.4, it can be clearly seen that our SDINet scheme shows comparative results when trained on the only individual type of input image.

Method	Precision (%)	Recall (%)	Fscore (%)
CTPN [25]	60.4*	53.8*	56.9*
SegLink [23]	42.3*	40.0*	40.8*
EAST [32]	78.7*	49.1*	60.4*
SLPR [33]	80.1*	70.1*	74.8*
CTD-TLOC [30]	77.4*	69.8*	73.4*
PSENet [27]	80.49*	78.13*	79.29*
SDINet scheme (ours)	81.78	79.43	80.59

Table 5.3: Performance comparison when trained on only SCUT-CTW 1500 dataset [30]. * indicates that the result is taken from [27]

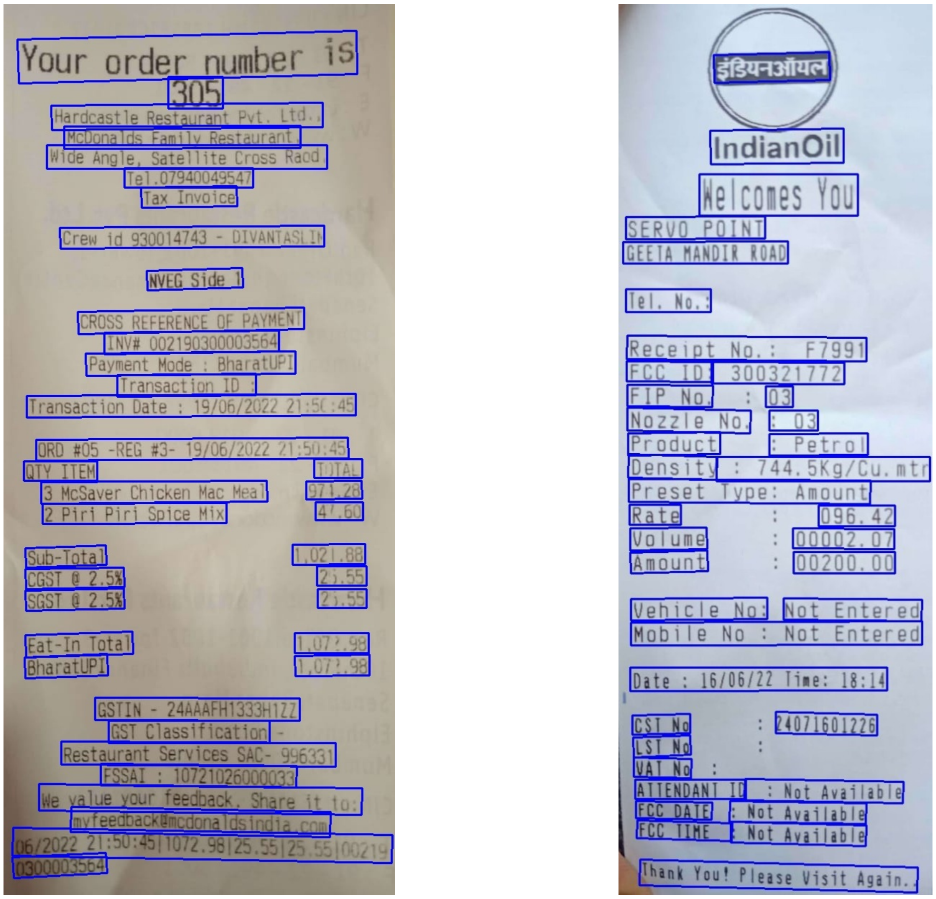
Method	Precision (%)	Recall (%)	Fscore (%)
Koo's [11]	68.53*	72.81*	70.61*
Fastext [1]	69.69*	81.89*	75.30*
EAST [32]	89.02*	92.75*	90.67*
RetinaNet [17]	86.78*	89.06*	87.91*
TextBoxes++ [14]	82.34*	88.79*	85.44*
DetectGAN [31]	98.98*	98.51	98.74
PSENet original [27]	96.70*	92.40*	94.34*
SDINet scheme (ours)	97.42	95.40	96.40

Table 5.4: Performance comparison when trained on only SROIE 2019 dataset [6]. * indicates that the result is taken from [31]

.

5.4 Detection Results Using the SDINet Scheme on Real Data

We collected real data from two different sources to see the detection result of SDINet scheme applied to the PSENet model on real collected data. The detection results obtained can be seen in Fig. 5.6. Fig. 5.6(a) shows the detection result obtained on the petrol pump bill collected from Indian Oil Corporation Ltd on 16 June 2022. Fig. 5.6(b) shows the detection result obtained on the invoice collected from McDonald's Family Restaurant on 19 June 2022. The result indicates that SDINet scheme is also performing well on real collected samples.



```
Your order number is
305
Hardcastle Restaurant Pvt. Ltd.,
McDonalds Family Restaurant,
Wide Angle, Satellite Cross Raod,
Tel.07940049547
Tax Invoice

Crew id 930014743 - DIVANTASLIN

NVEG Side 1

CROSS REFERENCE OF PAYMENT
INV# 002190300003564
Payment Mode : BharatUPI
Transaction ID :
Transaction Date : 19/06/2022 21:50:45

ORD #05 -REG #3- 19/06/2022 21:50:45
QTY ITEM                          TOTAL
 3 McSaver Chicken Mac Meal      971.28
 2 Piri Piri Spice Mix            47.60

Sub-Total                      1,021.88
CGST @ 2.5%                       25.55
SGST @ 2.5%                       25.55

Eat-In Total                   1,072.98
BharatUPI                      1,072.98

GSTIN - 24AAAFH1333H1ZZ
GST Classification -
Restaurant Services SAC- 996331
FSSAI : 10721026000033
We value your feedback. Share it to:
myfeedback@mcdonaldsindia.com
06/2022 21:50:45|1072.98|25.55|25.55|00219
0300003564
```

(a)

```
इंडियनऑयल
IndianOil
Welcomes You
SERVO POINT
GEETA MANDIR ROAD

Tel. No.:

Receipt No.:  F7991
FCC ID: 300321772
FIP No.   : 03
Nozzle No.  : 03
Product     : Petrol
Density : 744.5Kg/Cu.mtr
Preset Type: Amount
Rate        :   096.42
Volume      : 00002.07
Amount      : 00200.00

Vehicle No: Not Entered
Mobile No : Not Entered

Date : 16/06/22 Time: 18:14

CST No        : 24071601226
LST No        :
VAT No  :
ATTENDANT ID  : Not Available
FCC DATE   : Not Available
FCC TIME   : Not Available

Thank You! Please Visit Again..
```

(b)

Figure 5.6: Detection results using the proposed SDINet scheme on real samples: (a) Invoice collected from McDonald's Family Restaurant; (b) Petrol pump bill collected from Indian Oil Corporation Limited.

5.5 More Experiment and Failure Case

For checking the impact of hyperparameters on the result, the model is trained again from scratch by setting α and β to 0.5 which indicates that we are giving equal importance to the complete text and shrunk text instance loss. Fig. 5.7 shows the comparative result obtained on different values of α and β.

From Fig. 5.7, it can be seen that model yields different results on different hyperparameters. In our case, giving equal importance to complete text and shrunk text instance increases the performance of the model. Also, under the influence of the same hyperparameters, the imposition of our SDINet scheme still yields a comparatively better result. Hence, correct tuning of α and β can contribute to an increase in performance.

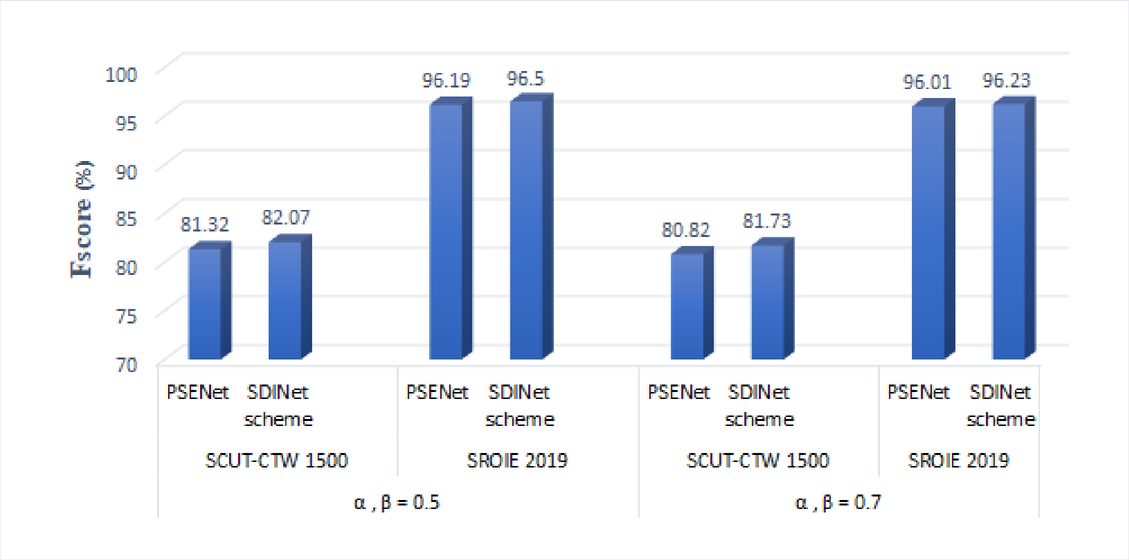


Figure 5.7: Results on different value of hyperparameters.

5.5.1 Failure Case

While the SDINet scheme can improve the performance of the generalized text detection model, there are still scenarios where it cannot separate the text instances that are so close to each as if they are composed within each other, see Fig 5.8. As shown in Fig. 5.8, the words MUDGEE and BREWING CO should have been separated, i.e. they should have been detected as different text instances rather than being detected as single text instances. The problem with such detection is twofold; one problem is the problem of missed detection where three text instances should have been extracted rather than two, and the second problem is the problem during the recognition stage when the very extracted text instance is supplied to recognition model it will include unnecessary noise within it resulting into the bad performance of recognition model. Our scheme does not do well in such a scenario because the decision about whether the neighbouring pixel is a part of the text instance or not becomes very complex. Hence, in such scenarios, a better network architecture is needed that can take better decisions about it.

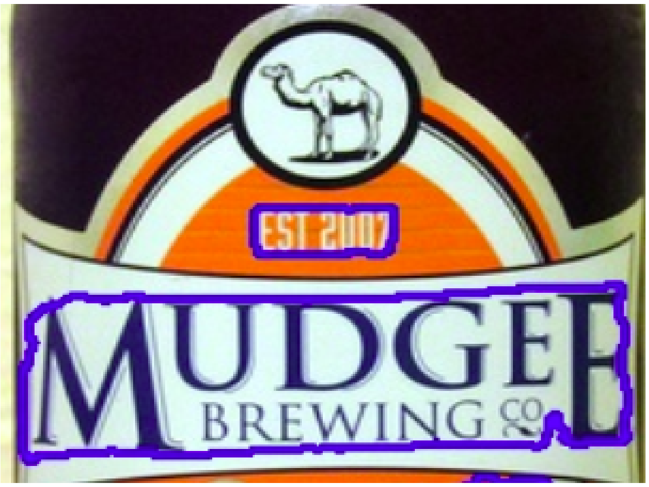


Figure 5.8: Words MUDGEE and BREWING CO is detected as single text instance.

5.6 Chapter Summary

This chapter implements the SDINet scheme to the PSENet model to detect text from scene or document text images. The proposed scheme is trained by supplying scene text and document text images. For comparison, both the models are trained by following the same training configuration and dataset. The comparison result shows that our proposed scheme is performing better as compared to the original PSENet model. Moreover, our proposed scheme also shows comparative results when trained on individual types of input images. In the end, we discuss failure cases.

6 Conclusions

In this thesis, an SDINet scheme is proposed for generalized text detection in scene and document images. During the training of the model, a Weighted Loss (WL) is designed to better update the training parameters according to the input image type. A classification model is designed that helps us to find the WL by classifying an input image as a scene text type image or document text type image. The novelty of our approach is in the fact that the training parameters of the model are updated according to the input image type. After training the classification model, it helps us find the critical weight factors. The proposed scheme is implemented and trained by supplying a combination of scene text and document text images for better generalization of both kinds of images. The comparison result shows that the SDINet scheme applied to the PSENet model improves the performance of the original PSENet model when the text needs to be detected from an arbitrary image. Moreover, our proposed text detection scheme also shows comparative results when trained on individual types of input images as compared to other text detection methods. In our method, there are scenarios where it cannot separate the text instances that are so close to each as if they are composed within each other. Hence, in such scenarios, a better network architecture is needed that can take proper decisions about neighbouring pixels being part of a particular text instance or not.

7 References

[1] M. Busta, L. Neumann, and J. Matas. Fastext: Efficient unconstrained scene text detector. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1206–1214, 2015.

[2] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. J. Wu, and A. Y. Ng. Text detection and character recognition in scene images with unsupervised feature learning. In *2011 International Conference on Document Analysis and Recognition*, pages 440–445. IEEE, 2011.

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

[4] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.

[5] W. Huang, Z. Lin, J. Yang, and J. Wang. Text localization in natural images using stroke feature transform and text covariance descriptors. In *Proceedings of the IEEE international conference on computer vision*, pages 1241–1248, 2013.

[6] Z. Huang, K. Chen, J. He, X. Bai, D. Karatzas, S. Lu, and C. Jawahar. Icdar2019 competition on scanned receipt ocr and information extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1516–1520. IEEE, 2019.

[7] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[8] N. Islam, Z. Islam, and N. Noor. A survey on optical character recognition system. *ITB Journal of Information and Communication Technology*, 12 2016.

[9] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[10] S. Jadon. A survey of loss functions for semantic segmentation. In *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–7. IEEE, 2020.

[11] H. I. Koo. Text-line detection in camera-captured document images using the state estimation of connected components. *IEEE Transactions on Image Processing*, 25(11):5358–5368, 2016.

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[13] J.-J. Lee, P.-H. Lee, S.-W. Lee, A. Yuille, and C. Koch. Adaboost for text detection in natural scene. In *2011 International conference on document analysis and recognition*, pages 429–434. IEEE, 2011.

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

[15] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu. Textboxes: A fast text detector with a single deep neural network. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[16] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[19] N. Mufti and S. A. A. Shah. Automatic number plate recognition: A detailed survey of relevant algorithms. *Sensors*, 21(9):3028, 2021.

[20] L. Neumann and J. Matas. A method for text localization and recognition in real-world images. In *Asian conference on computer vision*, pages 770–783. Springer, 2010.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[22] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[23] B. Shi, X. Bai, and S. Belongie. Detecting oriented text in natural images by linking segments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2550–2558, 2017.

[24] 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 transactions on pattern analysis and machine intelligence*, 39(11):2298–2304, 2016.

[25] 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*, pages 56–72. Springer, 2016.

[26] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *2011 International conference on computer vision*, pages 1457–1464. IEEE, 2011.

[27] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, and S. Shao. Shape robust text detection with progressive scale expansion network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9336–9345, 2019.

[28] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu. Detecting texts of arbitrary orientations in natural images. In *2012 IEEE conference on computer vision and pattern recognition*, pages 1083–1090. IEEE, 2012.

[29] C. Yi and Y. Tian. Text string detection from natural scenes by structure-based partition and grouping. *IEEE Transactions on Image Processing*, 20(9):2594–2605, 2011.

[30] L. Yuliang, J. Lianwen, Z. Shuaitao, and Z. Sheng. Detecting curve text in the wild: New dataset and new solution. *arXiv preprint arXiv:1712.02170*, 2017.

[31] J. Zhao, Y. Wang, B. Xiao, C. Shi, F. Jia, and C. Wang. Detectgan: Gan-based text detector for camera-captured document images. *International Journal on Document Analysis and Recognition (IJDAR)*, 23(4):267–277, 2020.

[32] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang. East: an efficient and accurate scene text detector. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5551–5560, 2017.

[33] Y. Zhu and J. Du. Sliding line point regression for shape robust scene text detection. In *2018 24th international conference on pattern recognition (ICPR)*, pages 3735–3740. IEEE, 2018.