Scene Text Detection and Recognition

A DISSERTATION

submitted in partial fulfillment of the requirements
for the award of the degree of

Master of Technology (Computer Science)

by

Saurav Dhara

ROLL NO. CS2228

Under the supervision of

Dr. Ujjwal Bhattacharya

Computer Vision and Pattern Recognition Unit

INDIAN STATISTICAL INSTITUTE

CERTIFICATE

This is to certify that the dissertation entitled **”Scene Text Detection and Recognition”** submitted by **Saurav Dhara** to the **Indian Statistical Institute, Kolkata**, in partial fulfillment of the requirements for the degree of **Master of Technology in Computer Science**, is an authentic and genuine record of the research work carried out by the candidate under my supervision and guidance. I affirm that the dissertation has met all the necessary requirements in accordance with the regulations of this institute.

Date: **Dr. Ujjwal Bhattacharya**

Computer Vision and Pattern Recognition Unit
Indian Statistical Institute

ACKNOWLEDGEMENT

I thank the Almighty first and foremost for giving me the fervor and excitement I needed to finish this study project. My supervisor, Dr. Ujjwal Bhattacharya of the Computer Vision and Pattern Recognition Unit at the Indian Statistical Institute, has my sincere gratitude for all of his encouragement and hard work in helping me fully understand the research field and for supporting me during the entire process of writing my M. Tech. dissertation. Significant feedback, constructive dialogue, and productive exchanges with the supervisors were particularly influential in shaping the final structure and caliber of the M. Tech. dissertation work.

Without my friends' enthusiastic encouragement, I could not have finished this thesis. My profound thanks to my parents for their love, blessings, and steadfast support. Last but not least, I express my gratitude to the All-Powerful God for giving me the inner strength, bravery, and wisdom I need to successfully finish my research assignment.

(Saurav Dhara)

DECLARATION

I, **Saurav Dhara**, with **Roll No. CS2228**, declare that the dissertation titled **"Scene Text Detection and Recognition"** is my original work for the **Master of Technology in Computer Science** degree at the **Indian Statistical Institute, Kolkata**.

I confirm that all the content in this report is my own work and any external sources have been properly cited. I understand that plagiarism or using others' work without acknowledgment is taken very seriously and will have consequences.

Saurav Dhara

Saurav Dhara

Roll No: CS2228

ABSTRACT

Deep learning methods have significantly reduced the difficulties related to multi-oriented text detection in recent scene text detection advances. The restrictions of conventional text representations, like horizontal boxes, rotated rectangles, or quadrangles, make it difficult to recognize curved writing. In order to tackle this problem, we provide a novel approach that uses instance-aware segmentation to identify irregular scene texts. Our method presents a semantic segmentation model that is led by attention and is intended to accurately label the weighted borders of text areas. Tests on multiple popular benchmarks show that, In contrast to cutting-edge techniques, our methodology delivers better performance on curved text datasets and maintains comparable results on multi-oriented text datasets.

Simultaneously, despite encouraging results in scene text detection, the complexity of the multi-stage pipelines used by present approaches sometimes causes them to fail in difficult settings. We offer a strong and simplified pipeline that uses a single neural network to predict words or text lines of variable quadrilateral forms and orientations in complete images, removing the need for needless intermediate steps. This simplicity makes it possible to concentrate on creating neural network designs and loss functions. Our examinations using reference datasets reveal that our suggested approach performs substantially superior to the majority advanced methods concerning precision and efficiency.

Keywords scene text detection, attention

Contents

List of Figures

Chapter 1

INTRODUCTION

1.1 Introduction

Computer vision and artificial intelligence researchers have focused a great deal of emphasis on scene text detection since it is widely utilized in text translation, autopilot, and picture and video retrieval. One of the most difficult jobs in many computer vision applications is scene text detection because of the variety of text sizes, shapes, textures, and complicated backgrounds. Over the last ten years, a plethora of text detection techniques have been put out, most of which mainly rely on manually created features to discern between text and non-text areas. However, those conventional methods don't ensure robust text detection; instead, they demand an extensive amount of feature engineering. The field of text detection in scenes has grown significantly with the aid of new deep learning methods.

Deep neural network-based text identification techniques may generally be divided into two groups. The first one predicts the offsets from text region suggestions or the corner positions of text instances by regressing horizontal boxes, oriented rectangles, or quadrilaterals, such as [15] [16] [12].

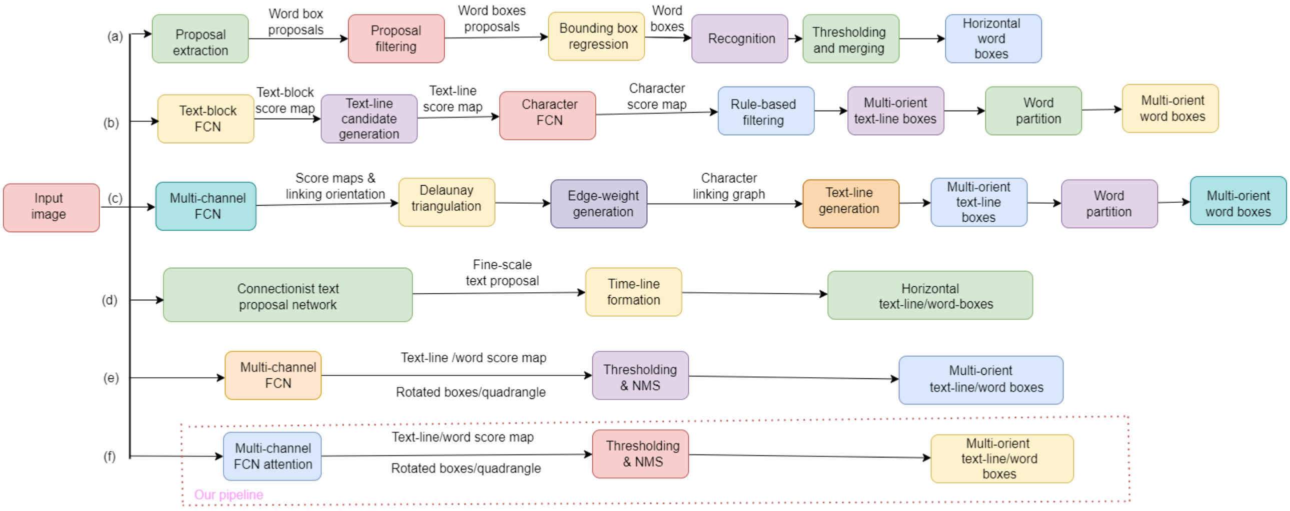


Figure 1.1: A comparison of some current efforts on scene text detection's pipelines:(a) The Jaderberg et al. [6] suggested horizontal word detection and recognition pipeline.(b) Zhang et al. [20] suggested multi-orient text detection pipeline.(c) Yao et al. [18] suggested multi-orient text detection pipeline ; (d) Tian et al. [16] proposed horizontal text detection utilizing CTPN.(e) Xinyu Zhou [21] proposed scene text detection.(f) our pipeline using attention mechanism

In this research, we present a two-stage, high-precision pipeline for scene text detection. The pipeline makes use of a fully convolutional network (FCN) model, which eliminates expensive and unnecessary intermediary steps to directly deliver word or text-line level predictions. To get the final results, the generated text predictions rotated rectangles are given to Non-Maximum Suppression. Additionally, we incorporate attention mechanisms [17] [2] into our model, such as channel and spatial attention modules, which significantly enhance scene text detection ability. Qualitative and quantitative studies on typical benchmarks show that the proposed approach delivers greatly improved performance while running much quicker than previous techniques.

The investigation makes the following conclusion:

- First, we suggest a two-stage approach for scene text detection: a fully convolutional network stage and an NMS merging stage. The FCN eliminates needless and time-consuming intermediate procedures and creates text areas immediately.

- Secondly, we incorporate attention modules to enhance detection performance. These modules help our model focus on the text regions, leading to a notable increase in detection accuracy.

- Third, the pipeline is versatile and can generate predictions at both the word and line levels, with output shapes that can be either rotated boxes or quadrangles, depending on the application's requirements.

- Fourth, we provide a fully trainable deep learning model end-to-end that outperforms existing state-of-the-art techniques when detecting text in scenes.

1.2 Motivation

In the modern digital age, the ability to automatically detect and recognize text in natural scenes is becoming increasingly vital. Scene text recognition is essential for a number of applications, including automatic license plate recognition, street sign reading for autonomous driving, and assisting visually impaired individuals in navigating their environment. Unlike traditional document text detection, scene text detection faces numerous challenges due to diverse backgrounds, varying lighting conditions, and different fonts and orientations of text.

Deep learning developments have created new avenues for tackling these issues. By leveraging powerful neural network architectures, we can create models capable of accurately identifying and localizing text in complex scenes. This project aims to push the boundaries of what is possible in scene text detection, making it more robust, efficient, and applicable to real-world scenarios. The ultimate goal is to enhance the accessibility of information embedded in everyday environments, thereby enriching user experiences and enabling innovative technologies.

1.3 Thesis Outline

The thesis is organized in the following way:

- First, we discuss some of the previous works concerned with similar problems and discuss their workings and limitations.
- Second, we discuss the methodology of our solution, where we discuss the different architectures of the models we used.
- Third, we discuss the dataset we used, how we preprocessed it, and the results of our experiments
- Lastly, we wrap up with a synopsis of our approaches along with their limitations. We also suggest some future research directions based on that.

Chapter 2

Related Work

Text detection from scenes has been a persistent area of research due to its critical applications in various fields such as document analysis, autonomous driving, and assistive technologies. Over the years, several approaches have been proposed to tackle the challenges posed by diverse text appearances and complex backgrounds in natural scenes such as stroke width transform (SWT) [3] it detects text segments in an image of a natural scene, It takes an RGB image as input and outputs a new image with the same dimensions that has the probable text highlighted, maximally stable extremal regions (MSER) [13] it is a method for detecting the points in images with different properties like brightness or color compared to the surroundings. Z. Zhang [19] utilized the local symmetry characteristic of text and developed a range of features for detecting text regions. FASText [1] a rapid technique for detecting text that was modified and enhanced the widely recognized FAST keypoint detector for stroke extraction. However, when it comes to accuracy and flexibility, these approaches pale in comparison to deep neural network-based ones, particularly in difficult situations where poor resolution and geometric distortion are present.

Deep neural network-based techniques have recently ushered in a new age for scene text identification [5,7,8] steadily taking the lead in the field. Huang [5] initially identified candidates using the MSER method and subsequently utilized a deep convolutional network as an effective classifier to eliminate false positives. Jaderberg [7] scanned the picture using a sliding-window method and using a convolutional model to produce a dense heatmap for each scale. After that, word candidates were identified by Jaderberg [6] using a CNN and an ACF combination. These were subsequently further improved using regression approaches. In order to recognize horizontal text lines, Tian [16] used a CNN-RNN hybrid model with vertical anchors. Zhang [20] suggested use an FCN [11] for heatmap creation and component projection for orientation estimate in contrast to these approaches. These techniques performed exceptionally well on common benchmarks. Xinyu Zhou [21] proposed a deep learning method for scene text detection that accurately forecasts phrases or text lines with quadrilateral forms and arbitrary orientations. It uses a fully convolutional network and combines loss functions for score maps and geometry, followed by non-maximum suppression for filtering detections. However, as Fig. 2(a-e) shows, these techniques usually include several steps and elements, including line construction, word partitioning, candidate aggregation, post-filtering for the elimination of false positives, and so on. The several steps and parts might require a lot of fine-tuning, which would lead to less-than-ideal performance and longer pipeline processing times.

FCN, with attention followed by NMS mechanisms, generally outperforms the EAST [21] and other scene text detection methods due to several reasons:

1. **FCN Based Pipeline:** This research presents a deep FCN-based architecture designed for text-level or word-level direct text identification. Our approach facilitates end-to-end training and

optimization by streamlining the process by eliminating needless intermediary components and processes, as seen in Fig. 1.1

2. **Enhanced Feature Representation:** Attention mechanisms allow the model to focus on relevant features while suppressing irrelevant ones. This leads to a more informative representation of text regions in the image, improving the accuracy of text detection, especially in cluttered scenes or with varying text sizes and orientations. **Adaptive Feature Selection:** Attention mechanisms dynamically weigh different parts of the image based on their importance for text detection. This adaptability allows the model to better handle diverse text appearances and complex backgrounds, leading to more robust text detection performance.

3. **Improved Contextual Modeling:** Attention mechanisms capture long-range dependencies and contextual information between different parts of the image. This contextual understanding helps the model to better distinguish text from the background and to accurately predict text regions, even when they are partially occluded or have irregular shapes.

4. **Reduced False Positives:** By focusing on relevant features and suppressing background noise, attention mechanisms can help reduce the number of false positive detections, improving the precision of the text detection system.

Overall, the integration of attention mechanisms into FCN enhances its ability to selectively focus on relevant information, adapt to different text appearances, and capture contextual relationships, leading to significant improvements in text detection accuracy and robustness.

Chapter 3

Methodology

Our suggested algorithm's main component is a neural network model that has been trained to identify text occurrences and their geometric characteristics from whole photos. This model generates detailed word or text line predictions at the per-pixel level. It is an FCN network specifically designed for text detection. It does this by doing away with intermediate steps such as candidate suggestion, text area building, and word partitioning. Only thresholding and NMS on anticipated geometric forms are included in the post-processing stages, which have been reduced.

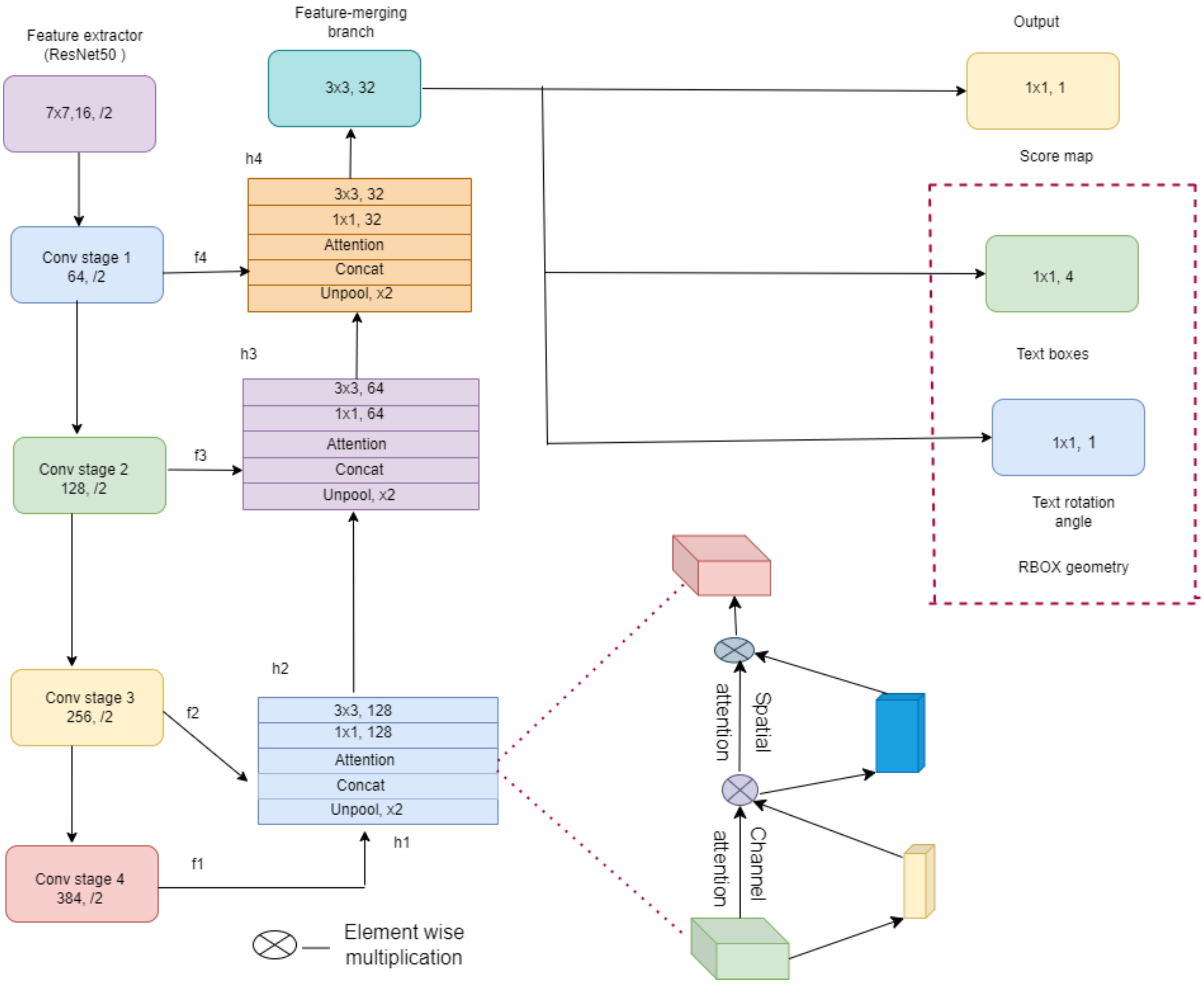


Figure 3.1: Structure of Detection FCN

3.1 Pipeline

Fig. 1.1 shows a high-level representation of our model architecture is given.Our model is based on DenseBox [4](high-level pipeline of densebox Fig. 3.2)

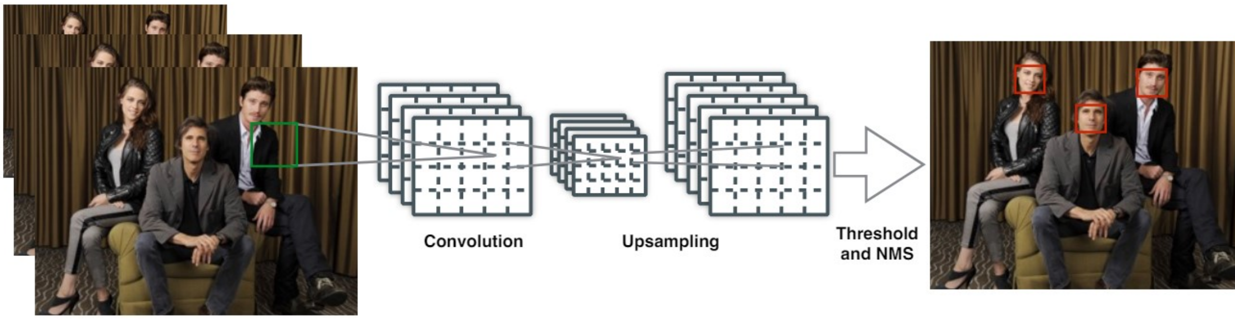


Figure 3.2: DenseBox Pipeline

which is a detection method where an image pyramid is processed by the network. To create the final result, the image is subjected to many layers of pooling and convolution, an upsampled feature map, and more convolution layers. After that, geometric data and text score maps at the pixel level are generated in numerous channels.

Pixel values on a score map, which has one output channel, range from 0 to 1. From the viewpoint of each pixel, the other channels depict the word's surrounding shapes. The anticipated geometric shape's degree of confidence at the associated place is indicated by the score.

Rotated Box (RBOX) text areas are a geometric form with which we have explored. We have conducted experiments with Rotated Box (RBOX) text sections, computing the loss of function on both the score map and on this geometry. Every projected area undergoes thresholding, keeping only geometries that have scores higher than a predetermined cutoff. Then, these geometries that are valid are stored for a later non-maximum suppression. The outcomes after NMS are regarded as the pipeline's ultimate output.

3.2 Network Architecture

We gradually integrate feature maps, using cues from designs such as Feature Pyramid Network (FPN) [10], and U-Net [14].Fig. 3.1 illustrates this strategy in pictorial form. Our main network for extracting features is ResNet50, and we use feature maps from its pooling layers 1 through 4. We use channel and spatial attention methods after the feature merging step.

Feature extractor network is pre-trained on ImageNet; extracted features are denoted by f_i.

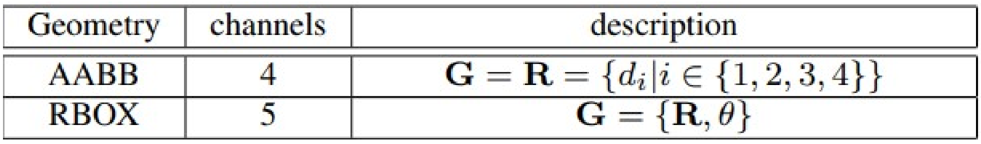
The feature merging process is defined by the following equations:

$$g_i = \begin{cases} \mathrm{unpool}(h_i) & \text{if } i \leq 3 \\ \mathrm{conv}_{3\times3}(h_i) & \text{if } i = 4 \end{cases} \tag{1}$$

$$h_i = \begin{cases} f_i & \text{if } i = 1 \\ \mathrm{conv}_{3\times3}(\mathrm{conv}_{1\times1}(\mathrm{A}_s(\mathrm{A}_c([g_{i-1}; f_i])))) & \text{otherwise} \end{cases} \tag{2}$$

where g_i is the merging base, h_i is the feature map after merging the spatial attention section is represented by A_s, and the channel attention section by A_c, $[;]$ denotes the concatenation operation. The feature map from the previous stage is supplied onto an unpooling layer in each merging step in order to double its size, and it is combined with the feature map from the current stage. Channel and spatial attention are then used.The output of this step is ultimately produced by a $conv_{3\times3}$, which fuses the information after a $conv_{1\times1}$ decreases the number of channels and processing. After the last step of merging, the last feature map of merging branch is created by a $conv_{3\times3}$ layer and sent to the resultant layer.

RBOX's geometry consists of a rotation angle θ and four axis-aligned bounding box (AABB) R channels.



Geometry	channels	description
AABB	4	$\mathbf{G} = \mathbf{R} = \{d_i \| i \in \{1,2,3,4\}\}$
RBOX	5	$\mathbf{G} = \{\mathbf{R}, \theta\}$

Figure 3.3: Geometry design output

The formula for R is the same as in [4], with four channels representing the separations between the pixel position and the rectangle's left, top, right, and bottom boundaries.

3.3 Attention Mechanism

Deep learning has advanced significantly with the use of the attention approach, particularly in applications that need the modeling of complicated data connections, such as image captioning, identification, and machine translation. This approach allows models to choose focus on certain areas of the incoming data, resulting in more efficient and accurate information processing.

3.3.1 Channel Attention

Traditional CNNs treat all channels on a feature map equally. To eliminate background interference, the channel attention mechanism assigns higher weights to channels that respond strongly to text areas. We initially perform average- and max-pooling procedures to input feature map, resulting in two distinct descriptors.The two descriptors are then sent through a shared network made up of single-layer perceptron.Finally adding these two output vectors element-wise and get the channel attention.

Mathematical Representation

The process can be mathematically expressed as follows:

$$M_c(f) = \sigma(\text{pool}_{avg}(f) + \text{pool}_{max}(f))$$

where σ is sigmoid actiavation function, pool_{avg} is average pooling and pool_{max} is max pooling operations.

3.3.2 Spatial Attention

A spatial attention module is a component in deep learning models, particularly in convolutional neural networks (CNNs), designed to enhance the model's ability to focus on important spatial regions within an image. This module highlights important regions and hides less useful ones in an effort to enhance performance on tasks like object identification, segmentation, and picture classification.

How It Works

The spatial attention module operates by generating an attention map that indicates the significance of different spatial locations in an input feature map. This map is then used to weight the input features, allowing the network to focus more on relevant areas. The process typically involves the following steps:

1. **Input Feature Map**: The module takes an input feature map, which is the output of a convolutional layer.

2. **Pooling Operations**: Two types of pooling operations, average pooling and max pooling, are applied across the channel dimension to capture different aspects of the features. This results in

two separate 2D feature maps:

$$\text{pool}_{avg}(f) : \text{Map of average pooled features}$$
$$\text{pool}_{max}(f) : \text{Map of max pooled features}$$

3. **Concatenation**: The pooled feature maps are concatenated to combine the information.

4. **Convolution**: A convolutional layer is applied to the concatenated feature maps to generate a spatial attention map. This map typically has a single channel.

5. **Activation Function**: The spatial attention map is passed through a sigmoid activation function (σ) to normalize the values between 0 and 1, which indicates the importance of each spatial location.

6. **Multiplication**: The normalized attention map is element-wise multiplied with the original input feature map to produce the final output, where important regions are emphasized.

Mathematical Representation

The process can be mathematically expressed as follows:

$$M_c(f) = \sigma(\text{Conv}_{7\times7}([\text{pool}_{avg}(f); \text{pool}_{max}(f)]))$$

Where:

- f is the input feature map.
- $\text{pool}_{avg}(f)$ is the average pooled feature map.
- $\text{pool}_{max}(f)$ is the max pooled feature map.
- $[\cdot; \cdot]$ denotes the concatenation operation.
- $\text{Conv}_{7\times7}$ is the convolution operation with kernel size 7.

- σ is the sigmoid activation function.

The channel attention and spatial attention modules are applied successively to perform the whole attention process given an intermediate feature map f.

3.4 Overview of the ResNet50 CNN architecture

ResNet-50 is a deep convolutional neural network (CNN) that is part of the Residual Networks (ResNet) family, introduced by Kaiming He et al. in 2015. ResNet-50 has become one of the most popular architectures due to its capacity to efficiently train extremely deep networks, introducing residual learning to solve the vanishing gradient issue. ResNet-50 consists of 50 layers and is known for its impressive performance on various image recognition tasks.

Key Features of ResNet-50

- **Residual Blocks**: The fundamental building blocks of ResNet-50 are the residual blocks. These blocks introduce shortcut connections (skip connections) that allow the network to learn residual functions instead of directly trying to learn unreferenced functions. This helps in mitigating the vanishing gradient problem.

- **Bottleneck Design**: ResNet-50 uses a bottleneck design for its residual blocks. A bottleneck block consists of three layers:

  - 1x1 convolution for reducing the dimensions
  - 3x3 convolution for processing the data
  - 1x1 convolution for restoring the dimensions

- **Identity and Convolutional Shortcut Connections**: There are two types of shortcut connections in ResNet-50:

  - Identity shortcuts that simply pass the input to the output without any modification.
  - Convolutional shortcuts that match the dimensions when the input and output dimensions differ.

- **Batch Normalization**: Each convolutional layer in ResNet-50 is followed by batch normalization and ReLU activation to ensure stable and faster training.

ResNet-50 Architecture

ResNet-50 is composed of the following layers:

1. **Initial Layers**:
   - 7x7 convolutional layer with 64 filters and a stride of 2 that is followed by ReLU activation and batch normalizing.
   - A 3x3 max pooling layer with a stride of 2.
2. **Residual Blocks**:
   - **Conv1**: 3 bottleneck blocks with 64, 64, and 256 filters respectively.
   - **Conv2**: 4 bottleneck blocks with 128, 128, and 512 filters respectively.
   - **Conv3**: 6 bottleneck blocks with 256, 256, and 1024 filters respectively.
   - **Conv4**: 3 bottleneck blocks with 512, 512, and 2048 filters respectively.
3. **Final Layers**:
   - A global average pooling layer.

- A fully connected (dense) layer with 1000 units (for classification into 1000 classes).

3.5 Loss Function

The loss function of the training is the sum of losses of score map and geometric loss which is defined as

$$\mathcal{L} = \mathcal{L}_s + \lambda \mathcal{L}_g \tag{3.1}$$

where $\mathcal{L}_s$ and $\mathcal{L}_g$ are losses corresponding to score map and geometric loss respectively. λ compares the significance of two losses.For our computation we have taken it's value as 1.

3.5.1 Score Map Loss

To address the uneven distribution of target objects, training pictures are meticulously processed using balanced sampling and hard negative mining in the majority of cutting-edge detection pipelines . It's possible that doing this might enhance network performance. Nevertheless, employing such methods invariably results in the introduction of a non-differentiable step, additional parameters to adjust, and a more intricate pipeline, which goes against our design premise.
A common approach is to use a weighted binary cross-entropy loss, where different weights are assigned to the text and non-text classes. The weighted binary cross-entropy loss can be defined as:

$$\mathcal{L}_s = -\beta Y \log(\hat{Y}) - (1-\beta)(1-Y)\log(1-\hat{Y})$$

where Y is the actual and $\hat{Y}$ is the prediction of score map.The compensating element for both positive and negative data, denoted by β, is defined by

$$\beta = 1 - \frac{\sum_{y \in Y} y}{|Y|}$$

3.5.2 Geometry Loss

One challenge to text recognition is the huge variability in font sizes observed in images of natural situations. Larger and longer text portions would favor a biased loss function in regression utilizing direct L1 or L2 loss. The regression loss must be scale-invariant as we must produce accurate text geometry predictions for both big and small text sections. Consequently, we use the IoU (intersction over union) loss in the RBOX regression's AABB section.
The Axis-Aligned Bounding Box (AABB) loss is defined as:

$$\mathcal{L}_{\text{AABB}} = -\log \text{IoU}(R_{\text{pred}}, R_{\text{act}}) = -\log \left(\frac{|R_{\text{pred}} \cap R_{\text{act}}|}{|R_{\text{pred}} \cup R_{\text{act}}|} \right)$$

where:

- $\text{IoU}(R_{\text{pred}}, R_{\text{act}})$ is the Intersection over Union between the predicted rectangle R_{pred} and the actual rectangle R_{act}.
- $|R_{\text{pred}} \cap R_{\text{act}}|$ is the area of the intersection between the predicted and actual rectangles.
- $|R_{\text{pred}} \cup R_{\text{act}}|$ is the area of the union of the predicted and actual rectangles.

If $\hat{d_1}$,$\hat{d_2}$,$\hat{d_3}$ and $\hat{d_4}$ represent the distances from a pixel to the top, right, bottom, and left boundaries of the predicted rectangle R_{pred}, and d_1,d_2,d_3 and d_4 represent the distances from a pixel to the top, right, bottom, and left boundaries of the actual rectangle R_{act} then then the width w_i and height h_i of the intersected rectangle $R_{\text{pred}} \cap R_{\text{act}}$

can be calculated as

$$w_i = \min(\hat{d}_2, d_2) + \min(\hat{d}_4, d_4)$$
$$h_i = \min(\hat{d}_1, d_1) + \min(\hat{d}_3, d_3)$$

then union is given by:

$$|R_{\text{pred}} \cup R_{\text{act}}| = |R_{\text{pred}}| + |R_{\text{act}}| - |R_{\text{pred}} \cap R_{\text{act}}|$$

Now the rotation angle loss is defined as

$$\mathcal{L}_\theta(\theta_{\text{pred}}, \theta_{\text{act}}) = 1 - \cos(\theta_{\text{pred}} - \theta_{\text{act}})$$

Last but not least, the weighted total of the rotation angle loss and the AABB loss represents the overall geometry loss.That is

$$\mathcal{L}_g = \mathcal{L}_{\text{AABB}} + \lambda_\theta \mathcal{L}_\theta$$

In our experiment I have taken $\lambda_\theta = 20$.

3.6 Training

ADAM [9] optimizer is used to train the model.Here I have used the input in size 512 × 512 × 3 and batch size 16. I set the learning rate at 0.001.Until the network's performance plateaus, it is trained.

3.7 Locality-Aware NMS

The proposed Locality-Aware Non-Maximum Suppression (LA-NMS) technique is made to deal with the copious amount of candidate geometries (bounding boxes) that dense predictions in text detection produce in an effective manner. Below is a summary of the main ideas:

3.7.1 Problem with Naïve NMS

Complexity: The temporal complexity of a typical NMS method is $O(n^2)$, where n is the number of possible geometries. Dealing with hundreds of thousands of candidates makes this unfeasible.

3.7.2 Proposed LA-NMS Approach

Locality Assumption: It is assumed that bounding boxes (geometries) from neighboring pixels have a strong correlation and may thus be handled locally.

Row-by-Row Merging: The method handles each geometry row by row as opposed to processing them all at once:

- **Iterative Merging:**Geometries are blended repeatedly inside each row. The total number of comparisons is decreased by merging the current geometry with the previous merged geometry.

3.7.3 Complexity Improvement

Best Case Scenario: In the best-case scenario, this row-wise and iterative merging approach reduces the complexity to $O(n)$.

Worst Case Scenario: In reality, the method performs efficiently since the locality requirement is often met, even though the worst-case complexity is still $O(n^2)$

3.7.4 Merging Process (WEIGHTEDMERGE)

Weighted Averaging: Two geometries (g and p) are merged by combining their coordinates according to their confidence scores:

- **Formula:** If $a = \text{WEIGHTEDMERGE}(g, p)$, then $a_i = V(g) \cdot g_i + V(p) \cdot p_i$, where a_i is a coordinate of a.

- **Score Calculation:** The score of the merged geometry $V(a)$ is the sum of the scores of g and p.

3.7.5 Key Differences from Standard NMS

Averaging vs. Selecting: This approach averages the geometries, so serving as a voting mechanism, as opposed to choosing one design based on the highest score. This is very helpful for video stabilization of detections.

Naming: Because of their comparable goal of eliminating redundant geometries, the terms "NMS" and "NMS" are still employed in functional description, despite these variances.

3.7.6 Summary of the Procedure

1. **Process Row by Row:**Instead of merging geometries all at once, merge them row by row.
2. **Iterative Merging:** Combine every geometry with the most recent one blended inside each row.
3. **Weighted Merge:**To stabilize the outcomes and raise the quality of detection, weight-average the coordinates of the geometries that are being merged using the scores.

3.7.7 Benefits

Efficiency: The algorithm operates more quickly in practice since the row-by-row method drastically lowers the amount of comparisons required.

Stabilization: For processing videos, weighted averaging yields a detection output that is more stable.

The LA-NMS algorithm successfully enhances text identification stability and performance in dense prediction settings by implementing these tactics.

Chapter 4

Experiments

To check the performance of the proposed method I experiment it on the datasets ICDAR 2015 and COCO-Text V2.0.

4.1 Benchmark Datasets

ICDR 2015

A benchmark for assessing text detection and identification algorithms in real-world scenarios is the ICDAR 2015 dataset, which is a component of the ICDAR 2015 Robust Reading Competition. For text localization, it contains 500 test pictures and 1000 training photos, all of which have text transcriptions and bounding boxes tagged on them. Issues including complicated backdrops, varying text appearances, and incidental text are addressed in the dataset. The ICDAR 2015 dataset, which offers a benchmark for evaluating and enhancing text detection and recognition systems in practical settings, is essential for the advancement of robust reading technologies.

COCO-Text V2.0

A large dataset called COCO-Text V2.0 is intended for text identification and detection in natural environments. With approximately 63,000 annotated text instances, bounding boxes, text transcriptions, and features including text type, language, and readability, it comprises 17,141 photos. The dataset tackles issues such as multilingualism, varied text appearances, and complicated origins. COCO-Text V2.0 is a vital resource for developing text detection and recognition technologies in practical settings. It is important for applications in text detection, text recognition, and end-to-end systems for automatic translation, scene understanding, and augmented reality.

4.2 Base Networks

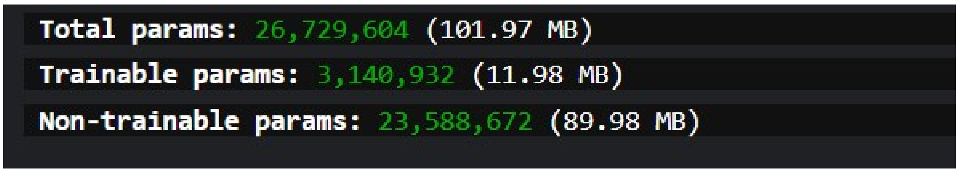
Since ICDAR 2015 is a tiny dataset, as I said, I trained my suggested model on two different types of datasets, therefore there is a risk of either over- or under-fitting. These two datasets were tested using several base models, including VGG16 and ResNet50. Additionally, contrast our approach with the network architecture model without focusing on it. Using the ImageNet dataset, I used pre-trained models VGG16 and ResNet50.

To compare model performance I have used mean-IoU metric.

4.3 Results

Attention Based Scene Text Dectection Model

Fig. 4.1 shows the total number of parameters are used in our model taking ResNet50 as base model



```
Total params: 26,729,604 (101.97 MB)
Trainable params: 3,140,932 (11.98 MB)
Non-trainable params: 23,588,672 (89.98 MB)
```

Figure 4.1: Attention Based Model Total Parameter

when I trained our model on ICDAR 2015 dataset the training loss of our model is given in figure Fig. 4.2

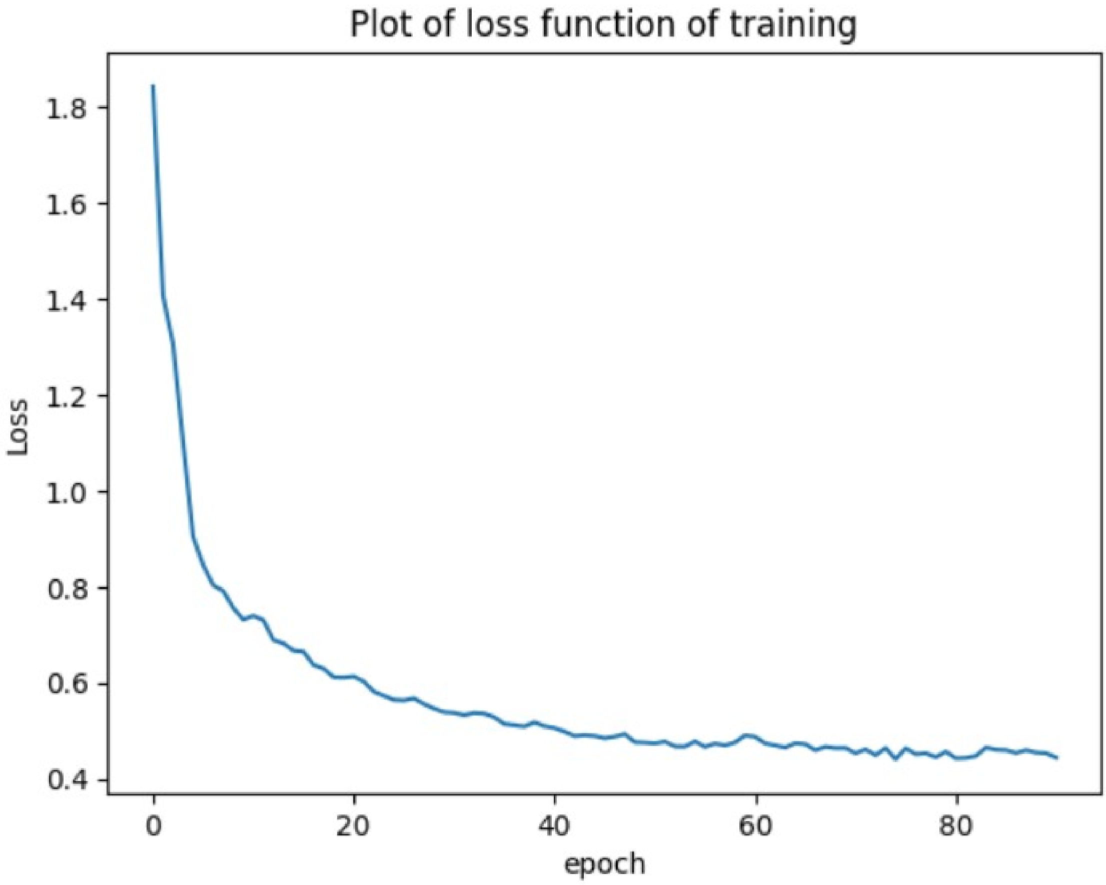


Figure 4.2: Training Loss on ICDAR 2015 Dataset

After apply the trained model on the test dataset Fig. 4.3 is some detection results

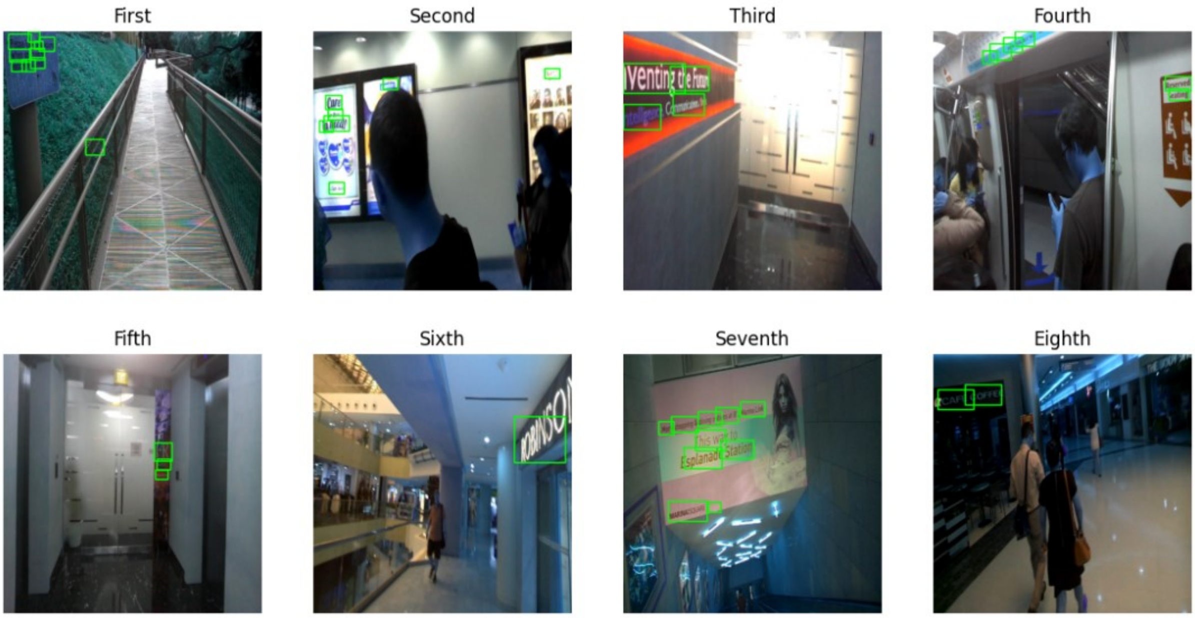
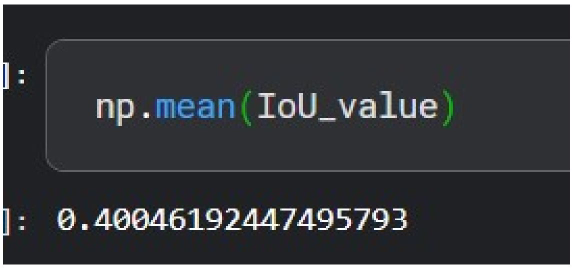


Figure 4.3: Resuls on ICDAR 2015 Test Dataset

and the performance metric mean-IoU on ICDAR 2015 test dataset is given in Fig. 4.4



```
]:  np.mean(IoU_value)

]:  0.40046192447495793
```

Figure 4.4: Performance on ICDAR 2015

The loss function graph after training on the COCO-Text V2.0 data is given in Fig. 4.5

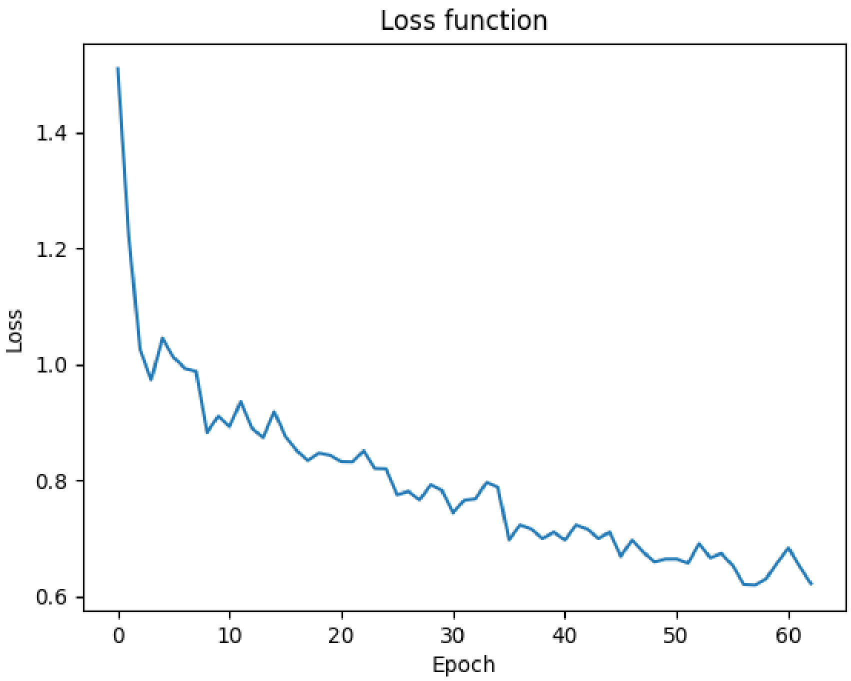


Figure 4.5: Training Loss on COCO-Text v2.0 Dataset

After apply the trained model on the test dataset Fig. 4.6 is some detection results

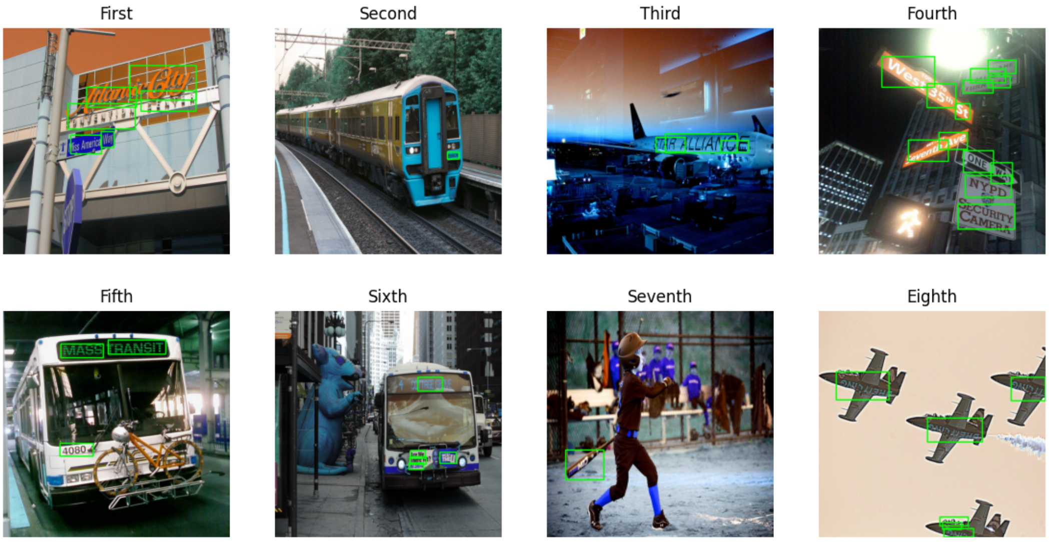
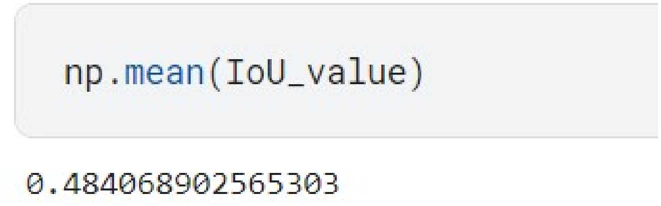


Figure 4.6: Resuls on COCO-Text v2.0 Test Dataset

and the performance metric mean-IoU on COCO-Text V2.0 test dataset is given in Fig. 4.7



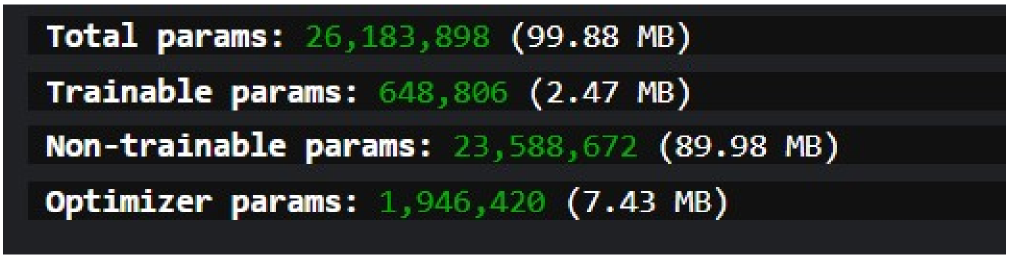
```
np.mean(IoU_value)

0.484068902565303
```

Figure 4.7: Performance on COCO-Text v2.0

Without Attention

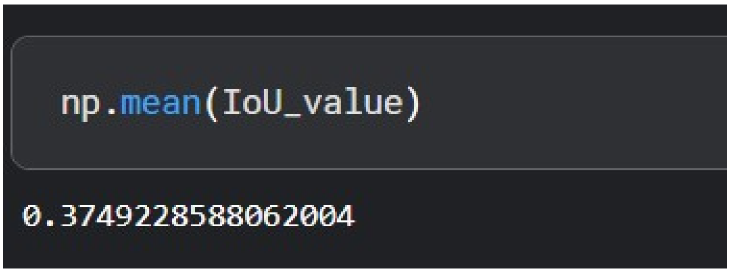
Taking ResNet50 as the base model the total number of parameters in the FCN structure is give by Fig. 4.8



```
Total params: 26,183,898 (99.88 MB)
Trainable params: 648,806 (2.47 MB)
Non-trainable params: 23,588,672 (89.98 MB)
Optimizer params: 1,946,420 (7.43 MB)
```

Figure 4.8: Without Attention model Total Parameter ResNet50 as base model

The performance on ICDAR 2015 dataset is given by Fig. 4.9

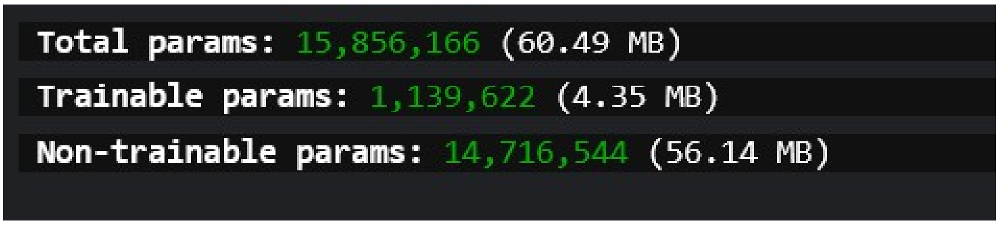


```
np.mean(IoU_value)

0.3749228588062004
```

Figure 4.9: Performance on ICDAR 2015 dataset

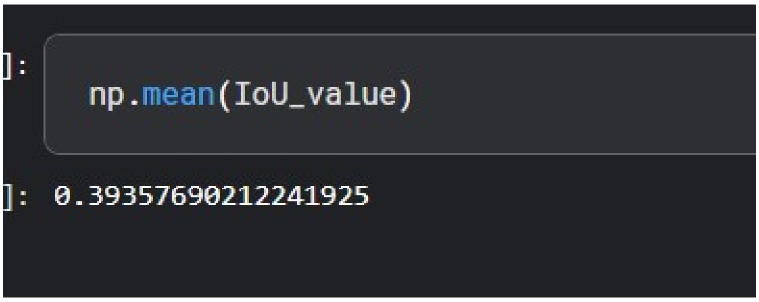
Taking Vgg16 as the base model the total number of parameters in the FCN structure is give by Fig. 4.10



```
Total params: 15,856,166 (60.49 MB)
Trainable params: 1,139,622 (4.35 MB)
Non-trainable params: 14,716,544 (56.14 MB)
```

Figure 4.10: Without Attention model Total Parameter Vgg16 as base model

The performance on ICDAR 2015 dataset is given by Fig. 4.11



```
]:  np.mean(IoU_value)

]:  0.39357690212241925
```

Figure 4.11: Performance on ICDAR 2015 dataset

Chapter 5

Conclusion

5.1 Limitation

Because the largest text instances that the detector can process efficiently are directly related to the receptive field of the network, the network is unable to accurately predict very long text sections, like lines of text that span the entire image, and because vertical text instances make up a small portion of the ICDAR 2015 training dataset, the algorithm may also be unable to detect or accurately predict vertical text instances.

5.2 Conclusion and Future Work

Conclusion

We have shown in this work that the model's performance for scene text identification is much improved by including an attention mechanism. at comparison to the baseline, the attention-enhanced model performed more accurately and robustly; it excelled at identifying texts in difficult contexts with a range of orientations and scales.

These advancements show how attention processes may be used to improve scene text identification algorithms and provide more accurate and consistent outcomes for uses like augmented reality and driverless driving.
For the recognition part I did't get much time to implement that part.

Future Work

Subsequent investigations may examine diverse attention pathways in order to enhance the precision and effectiveness of detection. Furthermore, our model's integration with cutting-edge OCR systems has the potential to provide a potent end-to-end text recognition pipeline. In order to enhance model generalization, there is also potential in utilizing bigger and more varied datasets and investigating semi-supervised learning strategies to make efficient use of unlabeled data. Important first steps toward a practical, broad implementation of the model will be to improve its resilience against occlusions and distortions and optimize it for real-time processing on mobile and embedded devices.

Bibliography

[1] Michal Busta, Lukas Neumann, and Jiri Matas. Fastext: Efficient unconstrained scene text detector. In *Proceedings of the IEEE international conference on computer vision*, pages 1206–1214, 2015.

[2] Jie Chen, Zhouhui Lian, Yizhi Wang, Yingmin Tang, and Jianguo Xiao. Irregular scene text detection via attention guided border labeling. *Science China Information Sciences*, 62:1–11, 2019.

[3] Boris Epshtein, Eyal Ofek, and Yonatan Wexler. Detecting text in natural scenes with stroke width transform. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 2963–2970. IEEE, 2010.

[4] Lichao Huang, Yi Yang, Yafeng Deng, and Yinan Yu. Densebox: Unifying landmark localization with end to end object detection. *arXiv preprint arXiv:1509.04874*, 2015.

[5] Weilin Huang, Yu Qiao, and Xiaoou Tang. Robust scene text detection with convolution neural network induced mser trees. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13*, pages 497–511. Springer, 2014.

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

[7] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Deep features for text spotting. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13*, pages 512–528. Springer, 2014.

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

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[10] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[11] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[12] Pengyuan Lyu, Cong Yao, Wenhao Wu, Shuicheng Yan, and Xiang Bai. Multi-oriented scene text detection via corner localization and region segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7553–7563, 2018.

[13] Lukas Neumann and Jiri Matas. A method for text localization and recognition in real-world images. In *Computer Vision–*

ACCV 2010: 10th Asian Conference on Computer Vision, Queenstown, New Zealand, November 8-12, 2010, Revised Selected Papers, Part III 10, pages 770–783. Springer, 2011.

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[15] Baoguang Shi, Xiang Bai, and Serge Belongie. Detecting oriented text in natural images by linking segments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[16] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image with connectionist text proposal network. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 56–72. Springer, 2016.

[17] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

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

[19] Zheng Zhang, Wei Shen, Cong Yao, and Xiang Bai. Symmetry-based text line detection in natural scenes. In *Proceedings of*

the IEEE conference on computer vision and pattern recognition, pages 2558–2567, 2015.

[20] Zheng Zhang, Chengquan Zhang, Wei Shen, Cong Yao, Wenyu Liu, and Xiang Bai. Multi-oriented text detection with fully convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4159–4167, 2016.

[21] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun 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.