Scene Text Detection

A DISSERTATION

submitted in partial fulfillment of the requirements

for the award of the degree of

Master of Technology (Computer Science)

by

Sugata Ghorai

ROLL NO. CS2232

Under the supervision of

Dr.Ujjwal Bhattacharya

Computer Vision and Pattern Recognition Unit

INDIAN STATISTICAL INSTITUTE

June 2024

CERTIFICATE

I hereby confirm that the research presented in the M.Tech. Dissertation entitled **Scene Text Detection**, submitted in partial fulfillment of the requirements for the degree of **Master of Technology (Computer Science)**, is a genuine record of my own work conducted from August 2023 to June 2024 under the guidance of **Ujjwal Bhattacharya,Computer Vision and Pattern Recognition Unit, Indian Statistical Institute.**

The matter presented in this thesis has not been submitted for the award of any other degree elsewhere.

Sugata Ghorai

Roll No. CS2232

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

12 Jun 2024

Ujjwal Bhattacharya

Computer Vision and Pattern Recognition Unit

Indian Statistical Institute

ACKNOWLEDGEMENT

I am grateful to the Divine for granting me the inspiration and drive to complete this research successfully. I extend my sincere appreciation to my supervisor, **Dr.Ujjwal Bhattacharya**, from the Computer Vision and Pattern Recognition Unit at the Indian Statistical Institute, for their continuous support, guidance, and encouragement throughout the duration of my M.Tech. dissertation. Their motivation and tireless efforts have been invaluable in gaining deep insights into the research area and shaping the quality of my work.

I am indebted to my friends whose support has been invaluable throughout this journey. My heartfelt thanks to my Parents for their unwavering love, blessings, and constant support.

Finally, I express my gratitude to the Almighty for granting me the strength, willingness, and wisdom to accomplish this research work successfully.

(Sugata Ghorai)

ABSTRACT

Scene text detection is crucial for numerous applications, including autonomous driving and assistive technology for visually impaired individuals. This project leverages various versions of the You Only Look Once (YOLO) model to achieve efficient and accurate text detection in natural scenes. Given YOLO's balance between speed and accuracy, it is an ideal candidate for real-time text detection tasks. Throughout this project, we compare different versions of YOLO, evaluating their performance on various multilingual datasets. These datasets comprise diverse scene text images with varying backgrounds, lighting conditions, and font styles. Each model is assessed based on metrics such as precision, recall, and mean Average Precision (mAP) score.As the YOLO versions are updated, their capability to detect text improves. Additionally, transfer learning is applied to datasets with common root languages, such as Hindi and Bengali or Telugu and Kannada. Our approach involves training these models in different ways and analyzing their performance on datasets with shared linguistic roots.Experimental results demonstrate that later YOLO versions significantly enhance text detection capabilities. This comparative analysis provides valuable insights into selecting the most suitable YOLO version for specific real-time text detection applications and highlights the benefits of transfer learning in multilingual contexts.

Contents

List of Figures

Chapter 1

INTRODUCTION

1.1 Introduction

Texts embedded in natural scenes, such as road traffic signs, billboards, and signage in shopping malls, play a crucial role in providing essential information about our surroundings.Real-time text detection [11] [10] [5] [6] and recognition [4] [22] is essential for many applications, such as autonomous robot navigation, intelligent driving systems, real-time text translation, and help for the blind. These applications demand not only high accuracy but also rapid processing speeds to avoid any delay that could lead to serious consequences.

Unlike standard text found in printed documents or on digital screens, scene text presents unique challenges due to its variability. Scene text can appear in multiple languages, sizes, fonts, colors, and orientations. Additionally, the text is often encountered under diverse lighting conditions, against complex backgrounds, and from various viewing angles. These factors contribute to the difficulty of accurately detecting and recognizing text in natural scenes.

In recent years, deep learning techniques have significantly advanced the field of object detection. Among these, the You Only Look Once (YOLO) [17] [18] family of models has gained popularity due to their impressive balance between detection speed and accuracy. YOLO models perform object detection in a single forward pass through the network, making them highly suitable for real-time applications.

All over the world more than seven thousands languages are spoken and almost three hundred languages are in written form. Most of the languages are region specific but they are of same ancestral script e.g. Hindi and Bengali are from Brahmi script or English and French are from Latin script. Now require datasets from all the languages are not available sufficiently to train a model.

Time and again, many state-of-the-art methods have emerged for text detection, but these typically require training on different languages, which poses a challenge due to the lack of comprehensive datasets for many local languages. Consequently, some languages are often overlooked, and many advanced models fail to detect finer texts within images. To address this issue, our approach experiments with related languages that share the same ancestral scripts. Additionally, we employ a Tiling Window Technique along with the YOLO models to capture fine details in the images, ensuring better detection accuracy for smaller text.
In this project, we leverage a variety of publicly available datasets such as MLe2e [3], ICDAR 2015 [8], and IIIT-ILST [16] to train, validate and test our model for different languages. Our approach involves using these datasets to ensure that our model can handle diverse linguistic contexts effectively.

To enhance the performance of our model, we sometimes employ transfer learning [19] techniques. This involves initially training the model on one group of datasets and then fine-tuning it on another set of languages by freezing some of the layers. This method allows us to retain the valuable features learned during the initial training phase while adapting the model to new languages with reduced computational effort.

We have compiled a comprehensive report that details the various models we have utilized in this project. This report includes a thorough comparison of the models, highlighting their strengths and weaknesses. As we analyze this report, we observe that larger models tend to be more effective in certain scenarios, demonstrating better performance due to their capacity to learn complex features. However, it is important to note that their effectiveness can vary depending on the specific use case.

Although the detection accuracy is not perfect in every instance, our models show promising results for real-time text detection. They are particularly effective in scenarios where quick and reliable text recognition is crucial. The findings from our report provide valuable insights into the trade-offs between model size, performance and detection accuracy, offering guidance for future improvements and applications in real-time text detection tasks.

1.2 Motivation

The motivation behind this work stems from the growing demand for effective text detection systems that can accurately process multilingual content in real-world applications. In today's interconnected world, digital content is produced and consumed in diverse languages, spanning geographical regions and cultural contexts. However, existing text detection models often struggle to adapt to the linguistic diversity and complexity encountered in real-world scenarios.

By developing robust and versatile text detection models capable of handling multilingual text, we aim to address this gap and unlock a wide range of applications across various domains. From document processing and translation services to image analysis and content moderation, accurate text detection is fundamental to enabling efficient information extraction and understanding.

Moreover, our work is motivated by the need to bridge the gap between research advancements in deep learning and practical applications in multilingual text detection. While significant progress has been made in the field of computer vision, there is still room for improvement, particularly concerning the development of models that can effectively handle diverse linguistic contexts.

By leveraging state-of-the-art techniques such as transfer learning and utilizing publicly available datasets, we aspire to contribute to the advancement of text detection technology. Our ultimate goal is to empower businesses, organizations, and individuals with the tools and solutions needed to process, analyze, and interpret multilingual text data accurately and efficiently.

Through this work, we aim to not only advance the state-of-the-art in text detection but also to facilitate broader access to multilingual content and promote

cross-cultural understanding in an increasingly interconnected world.

1.3 Approach

Our approach involves collecting diverse multilingual text datasets, including MLe2e, ICDAR 2015 and IIIT-ILST and preprocessing them for standardization. We then explore various versions off deep learning, prioritizing models that balance computational efficiency with multilingual performance. After partitioning datasets into training, validation, and test sets, we train the selected model, optimize parameters, and utilize transfer learning techniques to adapt the model to different language groups, reducing the need for extensive training data. Evaluation involves assessing model performance using standard metrics and qualitative analysis for real-world applicability, followed by comparison and analysis to identify areas for improvement. Finally, we compile a concise report summarizing our approach, experimental findings, and insights to guide future research in multilingual text detection.

Chapter 2

RELATED WORKS

Over the past few decades, scene text detection and recognition have been significant research areas within computer vision. Extensive surveys and in-depth analyses have been conducted on this topic [12] [13]. Traditional methods for detecting text in natural scenes heavily depend on handcrafted features to differentiate between text and non-text elements. Techniques such as sliding window (SW) [15] [14] and connected component (CC) [20] methods are commonly used .Using a pre-trained classifier, SW methods move a multi-scale detection window across all feasible positions inside an image and decide if the window includes text. But this method produces a lot of redundant detection windows, which makes it much less effective. Notable CC-based algorithms are Stroke Width Transform (SWT) [1] and Maximum Stable Extreme Regions (MSER) [5], which recognise connected components as possible characters that are subsequently categorised as text or non-text. The MSER technique has demonstrated good performance in contests like ICDAR2015 and ICDAR2013 [9]. Despite this, traditional methods fall short in terms of accuracy and flexibility compared to deep neural network-based methods, especially in challenging scenes involving low spatial resolution and geometric distortions.

There are two types of bounding box regression methods: one-stage and two-stage. Region proposals are essential to two-stage algorithms, of which Faster R-CNN [2] is the most prominent example. Modern techniques such as RRPN [21] and R2CNN [7] are based on Faster R-CNN. The axis-aligned region suggestions produced by RPN are subjected to numerous instances of Region-of-Interest (RoI) Pooling with varying

pooling sizes in R2CNN . The concatenated features are then utilised to categorise the proposals, estimating both inclined and axis-aligned region boxes. R2CNN can't perform multi-scale detection jobs, though; it only employs one detection scale. In order to detect rotated text, the RRPN technique extends the RoI pooling layer to a rotation RoI pooling layer by including rotation into the region proposal network. A two-stage detection system based on a Scale-based Region Proposal Network (SRPN) was presented by another method . Text scale estimate, text border determination, and text and non-text categorization are all part of the first step. Within the first stage's text region proposals, a detector in the second stage forecasts text boundary boxes. On the other hand, one-stage approaches do not depend on region suggestions and instead estimate candidate targets directly.

The networks YOLOv3 , YOLOv4 , and Single Shot Multibox Detector (SSD) are typical for one-stage techniques. Inspired by SSD , TextBoxes++ is a fast, end-to-end scene text detector that uses a single deep neural network. TextBoxes++ enhances the receptive field to encompass extended text regions by predicting bounding boxes using "long" convolution kernels. Cascaded NMS is used in testing to overcome the time-consuming nature of regular NMS computations. However, in dense text sections, cascaded NMS may reduce errors because it ignores the angle and distance of inclined bounding boxes. A single-shot text detector was proposed by He et al. that uses an attention strategy to reduce background interference in the convolutional features by highlighting text locations in images.Regression and classification branches are used in the RRD [20] approach for text detection feature extraction. By rotating convolutional filters, the regression branch extracts features that are sensitive to rotation, while the classification branch pools these features to produce features that are invariant to rotation. Dual-branch feature extraction, however, is less appropriate for real-time applications since it requires a lot of computing power and provides only modest accuracy gains.

One-stage approaches do not rely on region proposals; instead, they regress the bounding box straight from the convolutional feature maps, in contrast to two-stage methods. Because of their increased computing efficiency, one-stage algorithms are

better suited for quick detection in real-time applications.

In this paper, we have conducted an extensive survey covering different versions of the YOLO (You Only Look Once) object detection models, providing a detailed comparison among them. Our analysis goes beyond merely presenting the performance metrics of each version; we delve into the architectural innovations and enhancements that contribute to their efficiency and accuracy.

Additionally, this project transcends the boundaries of a single programming language. We implemented the models across multiple languages to explore the versatility and adaptability of YOLO in various computational environments. This multilingual approach allowed us to verify the models' robustness and ease of integration into diverse technological ecosystems.

A key focus of our research was the importance of transfer learning, particularly among languages with common ancestral roots. By applying transfer learning techniques, we examined how pretrained models could be fine-tuned to enhance performance across related languages. This aspect of our study underscores the potential for leveraging existing models to reduce training time and computational resources while maintaining high accuracy.

Through our comparative analysis and multilingual implementation, we aim to provide a holistic understanding of the YOLO models' capabilities and their applicability in real-world scenarios. Our findings highlight the practicality of transfer learning in improving model efficiency and effectiveness across different linguistic and technical landscapes.

Another key focus of our study is to capture finer details from the scene. To achieve this, we employ the Tiling Window concept. This approach involves running the pretrained model on the image by dividing it into smaller subparts, or windows. This technique allows the model to analyze and detect more intricate details within each segmented portion of the image.

Chapter 3

METHODOLOGY

In this section, we provide an in-depth analysis of four different variants of the YOLO (You Only Look Once) architecture: YOLOv1, YOLOv3, YOLOv5, and YOLOv8. Each of these models represents an evolution in the YOLO family, offering improvements in accuracy, speed, and performance. Alongside these models, we will also delve into an enhanced version of YOLOv1 that incorporates an attention mechanism, which aims to improve its capability in focusing on relevant parts of the image.

We will thoroughly discuss the unique features and architectural differences of each YOLO version. This includes an examination of their respective loss functions, which are critical for training the models to detect objects accurately. The loss functions help the models learn to minimize the error in predicting bounding boxes and class probabilities.

To address the challenge of detecting fine text within scenes, we introduce and describe the Tiling Window algorithm. This method involves dividing an image into smaller, manageable sub-parts, which allows the model to focus on and accurately detect small and fine details that might otherwise be missed in larger, more complex images.

Finally, we will cover the concept of transfer learning, explaining how pre-trained models can be fine-tuned for specific tasks. This approach leverages existing knowledge from models trained on large datasets, making it easier and faster to train models on new tasks with potentially limited data. With the help of transfer learn-

ing we try to show how similar type text(text from same acestral lineage) can be detected quickly and how to detect texts in low light also.

Through these discussions, we aim to provide a comprehensive understanding of the advancements in YOLO architectures, the enhancements in detecting fine details, and the practical applications of transfer learning in improving model performance.

3.1 YOLOv1

YOLOv1 stands out as an exceptionally fast object detection technique, capable of processing images in real-time at forty-five frames per second. YOLO, short for "You Only Look Once," is characterized by its simplicity. It employs a single convolutional network that predicts multiple bounding boxes and their associated class probabilities simultaneously.

In contrast to other detection methods that use a sliding window approach to extract features, YOLO takes in the entire image during both training and testing phases. This enables YOLO to encode contextual information about classes as well as their appearance. Compared to Fast R-CNN, YOLO demonstrates significantly fewer background errors.

One of the strengths of YOLO is its ability to learn generalizable representations of objects. Even when trained on natural images, YOLO performs impressively well on artwork during testing, outperforming top detection methods like DPM and R-CNN by a significant margin.

3.1.1 Architecture

Other detection techniques rely on a sliding window approach to extract features, whereas YOLO processes the entire image during both training and testing phases. This allows YOLO to incorporate contextual information about classes and their appearances. In comparison to Fast R-CNN, YOLO demonstrates significantly fewer background errors.

YOLO learns versatile representations of objects. Even when trained on natural images and evaluated on artwork, YOLO outperforms leading detection methods like DPM and R-CNN by a considerable margin.

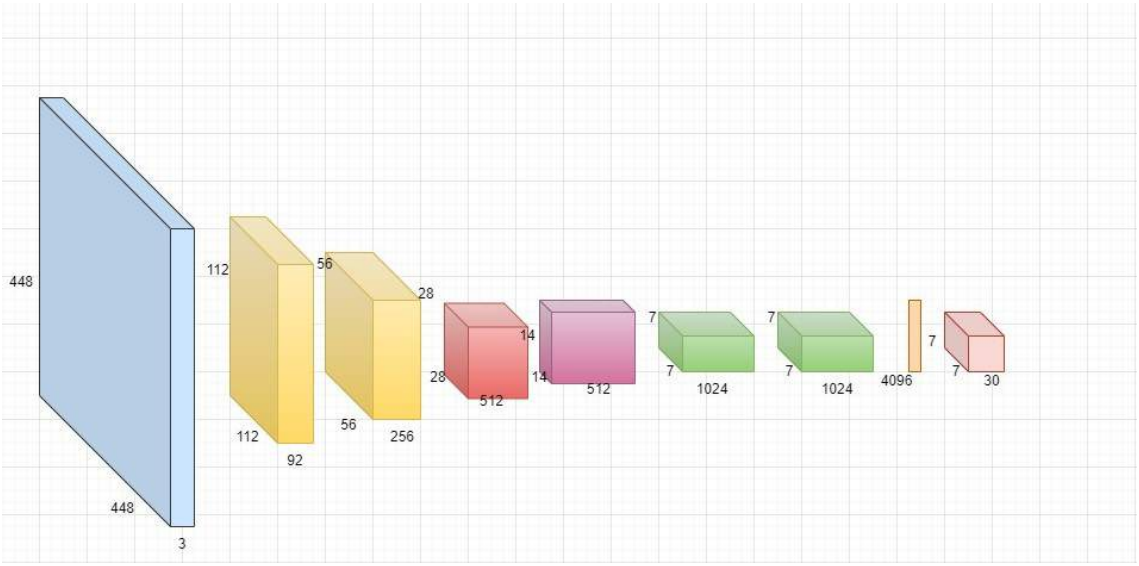


Figure 3.1: YOLOv1 architecture

The initial convolutional layers of the network extract features from the image, while the fully connected layers predict output probabilities and coordinates. YOLO consists of twenty-four convolutional layers followed by 2 fully connected layers.

3.2 YOLOv3

YOLOv3 (You Only Look Once, version 3) is an advanced real-time object detection system developed by Joseph Redmon and Ali Farhadi. It is known for its efficiency and accuracy, making it suitable for a wide range of computer vision tasks. Here are the main aspects of YOLOv3 architecture:

3.2.1 Backbone Network (Darknet-53):

- The backbone network used in YOLOv3 is Darknet-53, which consists of 53 convolutional layer
- It employs a combination of 3x3 and 1x1 convolutional layers along with residual connections, similar to those in ResNet.
- This deep and powerful network significantly improves feature extraction compared to earlier versions.

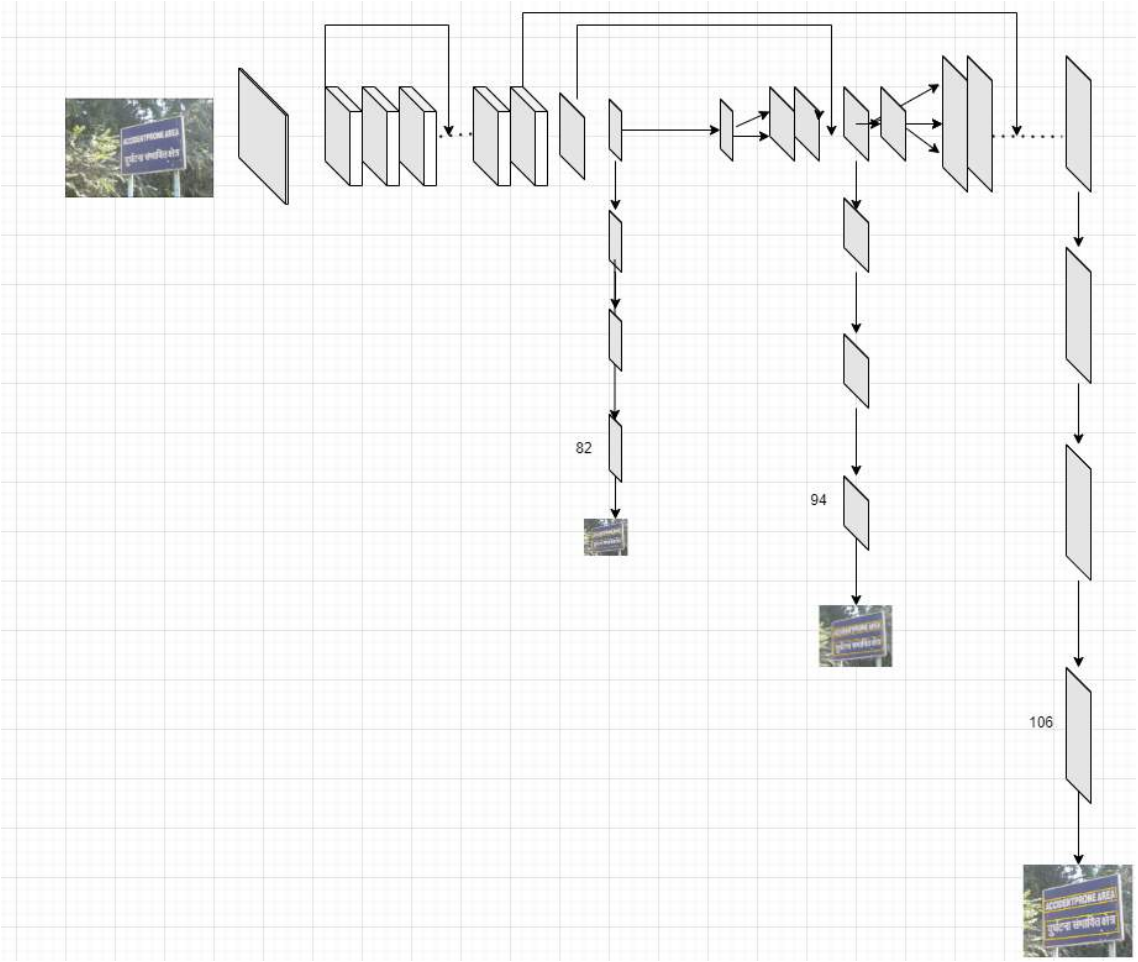


Figure 3.2: YOLOv3 architecture

3.2.2 Detection Head

- YOLOv3 detects objects at three different scales to effectively handle objects of varying sizes.
- Detection is performed using 1x1 detection kernels applied to feature maps from the backbone at three distinct layers.
- This multi-scale approach enhances the network's ability to detect smaller objects by leveraging finer details.

3.2.3 Anchor Box

- Predefined anchor boxes are used in YOLOv3 for predicting bounding boxes.
- During training, with the help of IoU score (Intesection over union)each ground truth box is matched with the most suitable anchor box.

- The network then predicts offsets to these anchor boxes, refining the bounding box predictions.

3.2.4 Feature Pyramid Network(FPN)

- To better detect small objects, YOLOv3 incorporates a structure similar to Feature Pyramid Networks (FPN) by using feature maps from different layers.
- This allows the network to utilize both low-level and high-level features, improving detection across various scales.

3.2.5 Prediction:

- At each scale, YOLOv3 predicts three boxes per cell, resulting in a total of nine predictions per grid cell.
- Each prediction includes coordinates (x, y, width, height), an score for objectness and class probabilities.

3.3 YOLOv5

YOLOv5 is a modern object detection model known for its optimized architecture, which significantly enhances both speed and accuracy. Here are the key components of the YOLOv5 architecture:

3.3.1 Backbone : CSPDarknet53

- CSPDarknet53: An advanced version of Darknet-53, integrating Cross Stage Partial (CSP) networks.
- CSP Networks: These networks split the feature map into two parts, process them separately, and then merge them. This improves gradient flow and reduces computational complexity, leading to more efficient feature extraction.

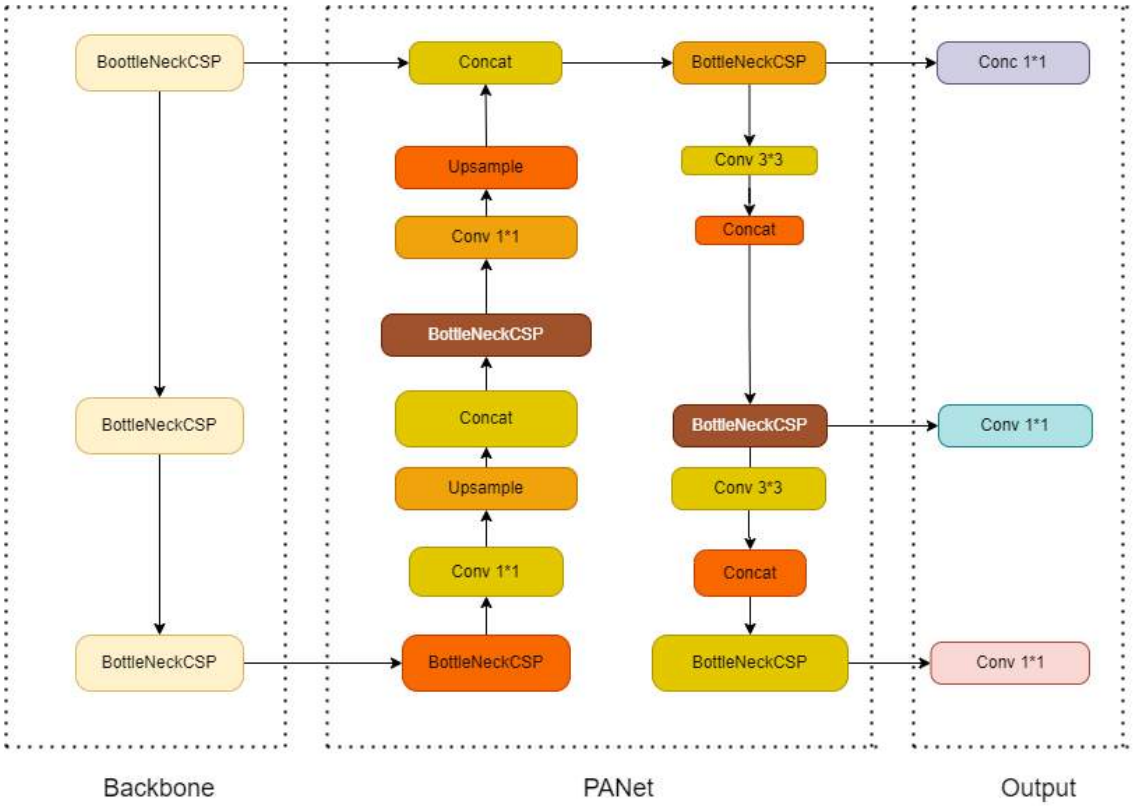


Figure 3.3: YOLOv5 architecture

3.3.2 Neck:PANet and FPN

- Path Aggregation Network (PANet): Enhances the feature pyramid by improving information flow from different levels of the backbone. This helps in better localization and detection of objects at multiple scales.

- Feature Pyramid Network (FPN): Facilitates the use of both low-level and high-level features from different stages of the backbone. This multi-scale approach is crucial for detecting objects of various sizes more effectively

3.3.3 Head: Detection Layers

- Detection Layers: These layers are responsible for predicting bounding boxes and class probabilities. YOLOv5 employs a more refined detection head compared to its predecessors, which improves the precision of bounding box regression and the accuracy of classification.

- Multi-Scale Predictions: The detection head makes predictions at three differ-

ent scales, allowing the model to handle objects of different sizes within the same image. Each scale outputs bounding box coordinates, objectness scores, and class probabilities.

3.4 YOLOv8

YOLOv8 stands out as a powerful object detection model that incorporates several advanced features like anchor free design and attention mechanism to boost its accuracy and efficiency.

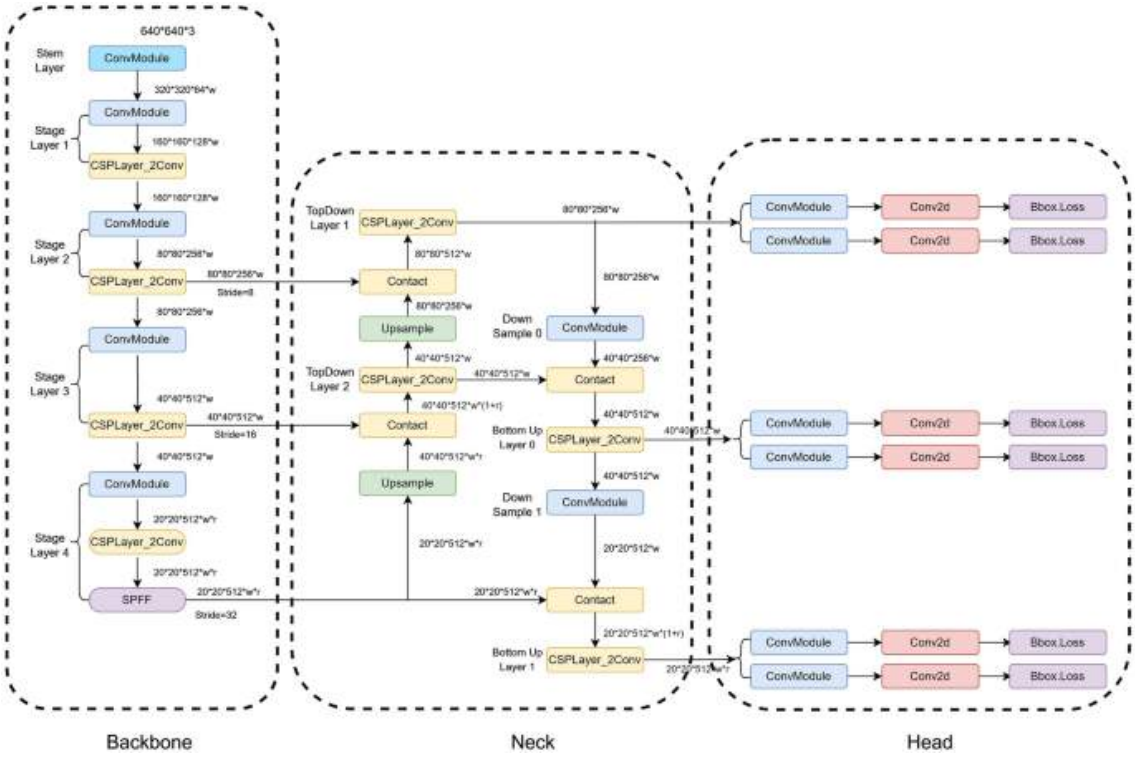


Figure 3.4: Architecture of YOLOv8

Architectural Overview

3.4.1 Input Layer:

- The input image is fed into the network, typically resized to a standard dimension (e.g., 640x640 pixels).

3.4.2 Backbone:

- The backbone network processes the input image through several convolutional layers, extracting hierarchical features. This backbone is often derived from popular architectures like CSPDarknet.

3.4.3 Neck:

- The Neck, which includes FPN and SPP layers, combines and refines features from different stages of the backbone. It ensures that multi-scale features are effectively utilized for object detection.

3.4.4 Head

- The detection head is responsible for predicting bounding boxes, objectness scores, and class probabilities. In YOLOv8, the anchor-free head simplifies the prediction process and improves efficiency.

3.4.5 Output:

- The final output consists of a set of bounding boxes with associated confidence scores and class labels. These predictions undergo post-processing to filter out low-confidence detections and merge overlapping boxes.

3.5 Loss Functions

The YOLO loss function consists of four components:

3.5.1 Localization Loss

The localization loss measures the accuracy of predicted bounding box coordinates compared to ground truth annotations. It is typically calculated using smooth L1 loss or mean squared error (MSE):

$$\text{Localization Loss} = \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right]$$

3.5.2 Confidence Loss

The confidence loss evaluates the model's confidence in object detection predictions using binary cross-entropy or sigmoid loss functions:

$$\begin{aligned}\text{Localization Loss} =& \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right. \\ & \left. + (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right]\end{aligned}$$

3.5.3 Classification Loss

The classification loss penalizes misclassifications using cross-entropy loss:

$$\text{Classification Loss} = \lambda_{\text{class}} \sum_{i=0}^{S^2} \mathbb{1}_{ij}^{obj} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

3.5.4 Bounding Box Prior Loss

The bounding box prior loss ensures predicted bounding boxes align with predefined anchor boxes:

$$\text{Bounding Box Prior Loss} = \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left[(w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2 \right]$$

Here, λ_{coord}, λ_{obj}, λ_{noobj}, and λ_{class} are hyperparameters controlling the importance of each loss component. S is the grid size, B is the number of bounding boxes per grid cell, x_i, y_i, w_i, and h_i are the predicted bounding box coordinates, C_i is the objectness score, $p_i(c)$ is the predicted probability of class c, and $\mathbb{1}$ is the indicator function.

3.6 Tiling Window

YOLOs are grat tool to detect the object(text). But often they are unable to produce to good results on tiny details. To detect the tiny details from the object we have used the concept of tiling windows and transfer learning together. Also to restrict the number of overlapping boxes we have used Non-Max Suppression technique(NMS).

3.6.1 Pseudo-code

Algorithm 1 Tiling window with NMS

Require: *image* (input image), *tile_size* (size of each tile), *overlap* (overlap between tiles)
Ensure: *tiles* (list of tiles with their positions)

```
height, width ← get_dimensions(image)
stride ← tile_size - overlap
tiles ← []
for y from 0 to height with step stride do
    for x from 0 to width with step stride do
        tile ← extract_subimage(image, x, y, tile_size, tile_size)
        if get_dimensions(tile) ≠ (tile_size, tile_size) then
            tile ← pad_tile_to_size(tile, tile_size, tile_size)
        end if
        tiles.append((tile, x, y))
    end for
end for
return tiles
{Function: get_dimensions(image)}
function get_dimensions(image):
    return (height of image, width of image)
{Function: extract_subimage(image, x, y, width, height)}
function extract_subimage(image, x, y, width, height):
    return image[y:y+height, x:x+width]
{Function: pad_tile_to_size(tile, target_width, target_height)}
function pad_tile_to_size(tile, target_width, target_height):
    current_height, current_width ← get_dimensions(tile)
    pad_bottom ← target_height - current_height
    pad_right ← target_width - current_width
    return pad_image(tile, pad_bottom, pad_right)
{Function: pad_image(image, pad_bottom, pad_right)}
function pad_image(image, pad_bottom, pad_right):
    return pad image with pad_bottom rows and pad_right columns of black pixels
```

Chapter 4

Dataset

Through out this project we have use several datasets commonly available over the internet. The dataset which we have used like MLe2e(multilingual end to end) , IIIT-ILST , ICDAR-2015 all are containing multilingual texts including English , Chinese , Korean , Hindi , Telugu , Kannada etc.

4.1 Data Preprocessing

- We first normalized the coordinates of the bounding box to make compatible with the input structure of the YOLO.
- Next we give the class labels as 'texts' for all the bounding boxes.
- The class label set as '0' in the input .
- **Example**
  -

0	0.88	0.15	0.33	0.14	0.34	0.27	0.88	0.28	0.88	0.15
0	0.88	0.31	0.33	0.30	0.31	0.42	0.88	0.41	0.88	0.31
0	0.72	0.43	0.33	0.45	0.35	0.56	0.72	0.53	0.72	0.43
0	0.69	0.56	0.36	0.58	0.35	0.73	0.71	0.69	0.69	0.56

- The first 0 is the class label , then we have four (x,y) coordinates of the bounding box

Figure 4.1: Scene text examples

Chapter 5

Experiments and Results

5.1 Metric

In this project we have used three following metric to validate the used models:

- Precision: Precision in object detection refers to the fraction of relevant instances among the retrieved instances. In simpler terms, it measures how accurate the model is when it detects objects. A high precision means that when the model predicts that an object is present, it is indeed correct most of the time.

- Recall: Recall in object detection refers to the fraction of relevant instances that have been retrieved over the total amount of relevant instances present in the dataset. It measures how well the model can find all the relevant objects. A high recall means that the model is able to detect most of the objects present in the image.

- mAP50:Mean Average Precision at IoU 0.5 (mAP50) is a widely used metric in object detection that assesses the overall effectiveness of a model across multiple object classes at a specific Intersection over Union (IoU) threshold of 0.5. This metric is particularly valuable as it integrates both precision and recall for all classes, offering a thorough evaluation of the model's performance in identifying objects.

5.2 Experiments

- First we train YOLOv1, YOLOv3, YOLOv5and YOLOv8 with MLe2e dataset.

Metric	YOLOv1	YOLOv3	YOLOv5	YOLOv8
Precision	0.67	0.75	0.83	0.91
Recall	0.59	0.74	0.80	0.88
mAP50	0.66	0.74	0.83	0.89

Table 5.1: Comparison of Evaluation Metrics Across YOLO Versions

- The followings are some outputs from the models.

Figure 5.1: Bounding boxes by YOLOs

- Then we add attention module in YOLOv3 and run the model on the same dataset MLe2e and get the followings:

Metric	Value
Precision	0.88
Recall	0.85
mAP50	0.88

Table 5.2: Performance Metrics of YOLOv3+attention

YOLOv3+attention

Figure 5.2: Bounding Boxes by YOLOv3+attention

- Since we have very few Hindi scene texts data so we have done two type of experiments: trained with pretrained model and another one is randomly initialized model.
- The results are given below.

Metric	Pretrained YOLOv3+attention	Non-Pretrained YOLOv3+attention
Precision	0.78	0.52
Recall	0.76	0.56
mAP50	0.77	0.51

Table 5.3: Comparison of Performance Metrics YOLOv3+attention

Metric	Pretrained YOLOv5	Non-Pretrained YOLOv5
Precision	0.75	0.58
Recall	0.73	0.51
mAP50	0.75	0.56

Table 5.4: Comparison of Performance Metrics YOLOv5

Metric	Pretrained YOLOv8	Non-Pretrained YOLOv8
Precision	0.83	0.66
Recall	0.82	0.64
mAP50	0.79	0.64

Table 5.5: Comparison of Performance Metrics YOLOv8

Figure 5.3: Bounding Box pretrained vs non pre-trained YOLOv5

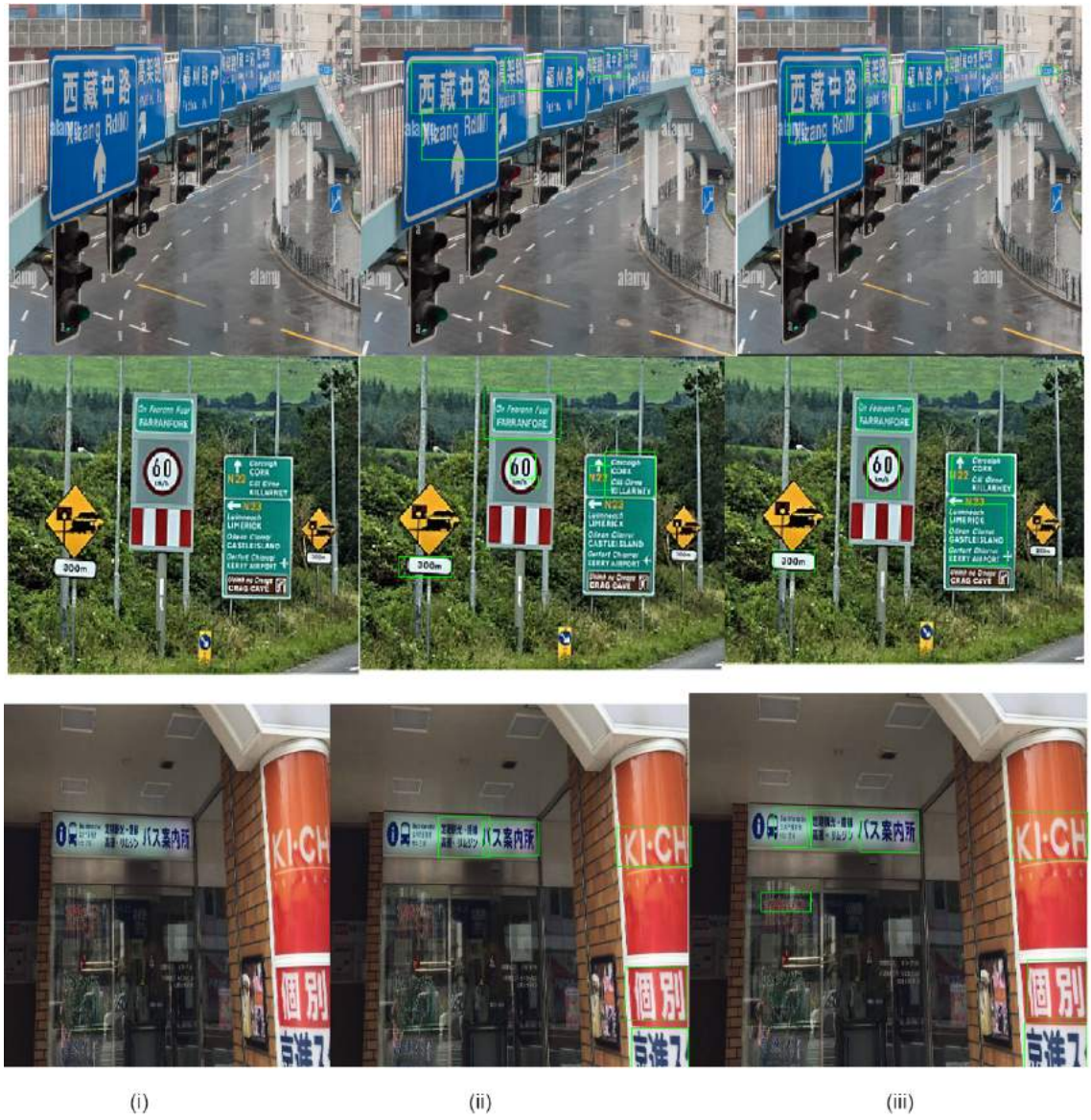
YOLOv8_trained

Yolov8_nontrained

Figure 5.4: Bounding Box pretrained vs non pre-trained YOLOv8

5.3 Effect of Tiling Window with Non-maximum Suppression(NMS)

- We have seen another drawback of the YOLO is that it can not detect small letters from a scene .
- Also it is not possible to annotate all the small texts on a scene .
- To overcome this we have used the concept of tiling window.



(i) (ii) (iii)

Figure 5.5: i)Original ii) Before Tiling Effect iii) After Tiling effect

5.4 Detection in Dim Light

- Our next target was to make our models compatible with different type of situation. The most crucial situation is detecting text on low light scenario.
- We are running out of that type of data.
- To overcome this case we train our pretrained model on some dim light images created from the existing images by adjusting the brightness .
- This make our model compatible with both type of situation.

- Presented below are several outputs generated by our trained models. These results illustrate the performance and effectiveness of our model

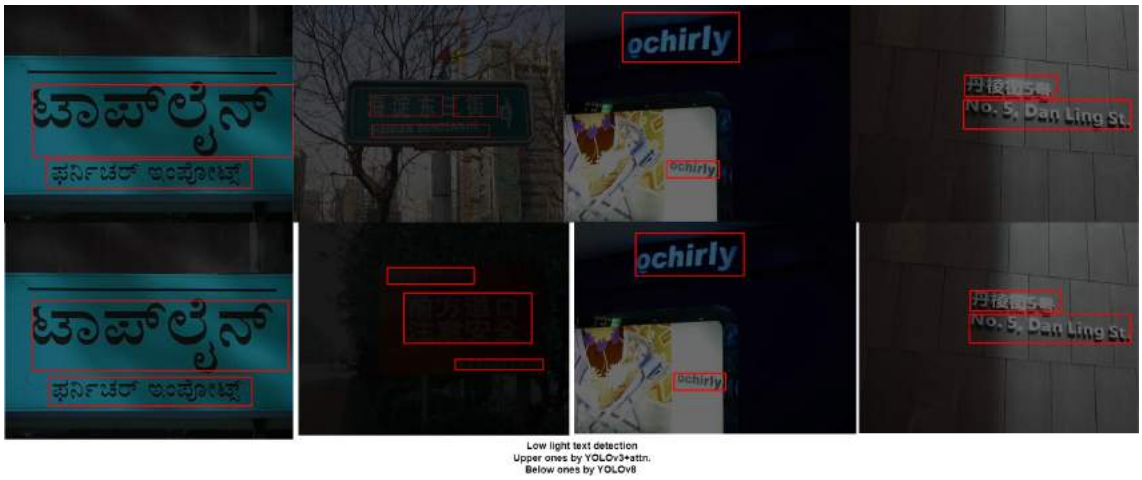


Figure 5.6: Bounding Boxes on dim light images

Chapter 6

Conclusion

In this study, we conducted an extensive evaluation of various YOLO models for multilingual text detection in images, encompassing languages such as English, Kannada, Chinese, Korean, Hindi, Telugu, and more. Our goal was to explore techniques to improve detection accuracy and robustness across different languages and diverse text styles.

We first trained and compared the performance of YOLOv1, YOLOv3, YOLOv5, and YOLOv8 architectures on the multilingual text detection task. Among these, YOLOv3 with integrated attention mechanisms emerged as the top performer, showcasing superior performance across all languages evaluated. Despite a slightly increased computational overhead, the attention mechanisms facilitated better localization of text instances across diverse languages and writing styles.

Furthermore, we enhanced the YOLOv3 architecture by incorporating attention mechanisms, which proved to be more effective than YOLOv5, particularly in handling multilingual text. The attention mechanism allowed the model to prioritize relevant regions, resulting in improved detection accuracy and robustness.

Through transfer learning, we demonstrated the effectiveness of pretrained models in capturing language-agnostic features and accelerating convergence across multiple languages. Pretrained models consistently outperformed models trained from scratch, underscoring the importance of leveraging pretrained weights for multilingual text detection tasks.

Our experiments consistently demonstrated that pretrained models outperformed

models trained from scratch in terms of both accuracy and convergence speed. This highlights the significance of leveraging pretrained weights for multilingual text detection tasks, as it enables the model to generalize better to unseen languages and reduces the need for large amounts of language-specific training data.

To refine text detection further, we employed tiling window and non-max suppression techniques. These methods improved the granularity of text localization and reduced false positives, particularly in complex scenes containing multilingual text.

Addressing challenges related to varying lighting conditions, we fine-tuned our pretrained models on datasets containing low-brightness images. This adaptation significantly improved detection performance under low-light conditions, showcasing the adaptability of our models across different environments.

In conclusion, our study demonstrates the effectiveness of attention mechanisms, transfer learning, and advanced post-processing techniques for multilingual text detection using YOLO models. These findings have significant implications for applications such as document analysis, scene understanding, and multilingual text recognition in real-world scenarios.

Chapter 7

Future Work

While our current study focused on text detection in multilingual images, there are several avenues for future research in text recognition:

7.1 Text Recognition

End-to-End Text Recognition: Investigate end-to-end text recognition models that directly recognize text from detected regions without relying on separate OCR systems. This approach can streamline the pipeline and improve efficiency.

Attention Mechanisms for Recognition: Explore the integration of attention mechanisms into text recognition models to improve their ability to handle long and irregular text sequences, especially in multilingual scenarios with varying text lengths and styles.

Language-Specific Models: Develop language-specific recognition models to capture linguistic nuances and improve recognition accuracy for each language. Fine-tuning pretrained models on language-specific datasets can be beneficial.

Multilingual Recognition: Design models capable of recognizing text in multiple languages simultaneously. Investigate techniques to handle code-switching and mixed-language text effectively.

7.2 Data Augmentation and Synthesis

Augmented Datasets: Generate augmented datasets with variations in font styles, sizes, orientations, and backgrounds for each supported language. This can improve the robustness of recognition models to different text styles.

Synthetic Data Generation: Explore synthetic data generation techniques, such as generative adversarial networks (GANs), to create large and diverse datasets for training recognition models, especially for languages with limited annotated data.

7.3 Cross-Modal Learning

Cross-Modal Fusion: Explore methods for integrating text detection and recognition modules into a unified framework, allowing joint learning and better integration of visual and textual information.

Multimodal Fusion: Investigate multimodal approaches that combine visual information from images with additional context (e.g., scene context, language context) to improve text recognition accuracy.

7.4 Robustness to Arbitrary Shapes

Handling Arbitrary Shapes: Investigate methods to make YOLO models robust to arbitrary shapes of text regions, such as curved or irregular shapes commonly found in calligraphy, artistic text, or scene text with perspective distortion.

Polygonal Bounding Boxes: Explore the use of polygonal or rotated bounding boxes instead of rectangular boxes to better fit non-rectangular text regions. This can improve the accuracy of text localization in scenes with complex layouts.

Instance Segmentation: Consider integrating instance segmentation techniques to precisely delineate text regions with irregular shapes. This can help in accurately recognizing text in challenging scenarios where text regions overlap or have complex shapes.

Bibliography

[1] Boris Epshtein, Eyal Ofek, and Yonatan Wexler. Detecting text in natural scenes with stroke width transform. pages 2963 – 2970, 07 2010.

[2] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[3] Lluis Gomez. Mle2e multi-lingual end-to-end dataset, 04 2016.

[4] Wenhao He, Xu-Yao Zhang, Fei Yin, Zhenbo Luo, Jean-Marc Ogier, and Cheng-Lin Liu. Realtime multi-scale scene text detection with scale-based region proposal network. *Pattern Recognition*, 98:107026, 2020.

[5] Weilin Huang, Zhe Lin, Jianchao Yang, and Jue 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] 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.

[7] Yingying Jiang, Xiangyu Zhu, Xiaobing Wang, Shuli Yang, Wei Li, Hua Wang, Pei Fu, and Zhenbo Luo. R2cnn: Rotational region cnn for orientation robust scene text detection, 2017.

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

[9] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. Icdar 2013 robust reading competition. In *2013 12th international conference on document analysis and recognition*, pages 1484–1493. IEEE, 2013.

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

[11] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[12] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11474–11481, 2020.

[13] Minghui Liao, Zhisheng Zou, Zhaoyi Wan, Cong Yao, and Xiang Bai. Real-time scene text detection with differentiable binarization and adaptive scale fusion. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):919–931, 2022.

[14] 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.

[15] Fagui Liu, Cheng Chen, Dian Gu, and Jingzhong Zheng. Ftpn: scene text detection with feature pyramid based text proposal network. *IEEE Access*, 7:44219–44228, 2019.

[16] M. Mathew, M. Jain, and C. V. Jawahar. Benchmarking scene text recognition in devanagari, telugu and malayalam. In *ICDAR MOCR Workshop*, 2017.

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

[18] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[19] Ricardo Ribani and Mauricio Marengoni. A survey of transfer learning for convolutional neural networks. In *2019 32nd SIBGRAPI conference on graphics, patterns and images tutorials (SIBGRAPI-T)*, pages 47–57. IEEE, 2019.

[20] Kamrul Hasan Talukder and Tania Mallick. Connected component based approach for text extraction from color image. In *2014 17th International Conference on Computer and Information Technology (ICCIT)*, pages 204–209, 2014.

[21] Thang Vu, Hyunjun Jang, Trung X. Pham, and Chang D. Yoo. Cascade rpn: Delving into high-quality region proposal network with adaptive convolution, 2019.

[22] Jian Ye, Zhe Chen, Juhua Liu, and Bo Du. Textfusenet: Scene text detection with richer fused features. In *IJCAI*, volume 20, pages 516–522, 2020.