



A report on

**TEXT LOCALIZATION AND SCRIPT
IDENTIFICATION USING YOLOv5**

Presented By:

Chandana, Apoorva, Anusha Upadhyaya, Naveen Patil

Contents

ABSTRACT	3
INTRODUCTION	4
APPLICATIONS	6
CHALLENGES	7
OBJECTIVES OF THE PROJECT	7
LITERATURE SURVEY	8
METHODOLOGY	10
A. DATA COLLECTION	10
B. YOLO MODEL ARCHITECTURE	12
C. PROPOSED MODEL	14
RESULTS	16
A. EXPERIMENTAL RESULTS	16
B. TENSORBOARD RESULTS	24
C. COMPARISON WITH OTHER METHODS	32
CONCLUSION	32
FUTURE SCOPE	33
REFERENCES	34

ABSTRACT

Script identification in color imagery and video content has gained popularity over the years in the field of research with an increase in demand for the extraction of textual information from natural scene images. In this paper, Text localization/ text detection and its corresponding script identification are performed on color images and videos. The texts are spotted in color images and videos and their scripts are distinguished in either English, Hindi, or Kannada language. For text detection and script recognition, the state-of-the-art method called YOLOv5 is used. YOLO is a real-time, single-shot object detection algorithm with better performance and higher accuracy than other object detection algorithms. The proposed model is trained with a custom dataset containing images and labels of all the text present in each and every image and the model is tested for different scenarios like different backgrounds, fonts, orientations, resolutions, and disturbances in the images to check its robustness. Finally, the proposed model is compared with other existing models/ algorithms for text detection and script identification in color imagery.

INTRODUCTION

Automatic script identification has become a popular research concern in multi-scripting environments in recent years. The competence to read the text in natural scenes is a highly-desirable feature in humanistic applications of the computer vision field. The text always contains some predominant information to help us understand the environment. Scene text often appears on product packaging, license plates, product invoices, billboards, and more. Reading text in an image has several applications such as a visual question-and-answer system and text translation from an image. Unfortunately, the text seen in images and videos has arbitrary grayscale values which are not always in black or white, and other problems like low resolution, variable size, and complex backgrounds make it difficult to accurately detect text and identify scripts in a scene.

As the text is considered to be the most expressive means of communication and can be entrenched into documents/images/videos and other multimedia content as a means of communicating information. So as to make it perceptible/readable by others. [2] The collection of massive amounts of street-view data is one such compelling application. The other factor that can be considered is the increasing availability of high-performance mobile devices, with both imaging and computational capability. This creates an opportunity for image acquisition and processing anytime, anywhere, making it suitable to recognize text in various environments. Lastly, the advances in computer vision and pattern recognition technologies along with deep learning, make it more feasible to address demanding problems.

Over the years, there were multiple solutions developed for the identified problem. One among them is Optical Character Recognition (OCR) which is viewed as a solved problem and is recommended for text detection and recognition. Though OCR performs well for document images it does not work efficiently for images/documents having complex backgrounds, arbitrary shaped texts, and different sizes of text. It also includes multiple processing steps, feature extraction, and filtering. This also requires a great deal of effort in fine-tuning the parameters and slows down the detection process. Still, there are abundant opportunities for further research to improve text detection accuracy.

In countries like India where people are accustomed to having many regional languages like Kannada, Hindi, Gujarati, Marathi, etc. along with English. It is seen that these languages are often used in natural scenes, street-view video/ image data, and in some documents. Among these languages most popularly used in India are Hindi, Kannada, and English. It is comparatively easy to detect and identify English texts as it consists of 26 lowercase or uppercase characters. While the complication arises with scripts like Kannada and Hindi which do not have lowercase or uppercase characters that are seen in the English language. Text localization and identification for such scripts become an arduous task with OCRs

Hindi is written in Devanagari scripts. [4] The Devanagari script comprises 13 vowels, 34 consonants, and 14 vowel modifiers. A whole letter is created by connecting a modifier with consonants. Therefore, the shapes of composite characters are more complex than consonants. Kannada script is similar to Hindi but differs in curves of the characters and often it is hard for the machine to differentiate these two scripts. Over the years, researchers have presented many models like [6] Fully-Convolutional Regression Network (FCRN), [4] Region Proposal Network (RPN) with feature pyramid network using ResNet, and [1] Geometrical Machine Learning Algorithm, etc. for text localization and detection of these scripts. Nevertheless, there is ample room to improve the accuracy and performance of text localization and script identification of these languages.

In this paper, we propose Text localization and Script identification of Hindi, English, and Kannada using YOLOv5. You only look once (YOLO) is a state-of-the-art object detection algorithm that uses Convolution Neural Network. YOLO uses darknet53 as the backbone. [10] It applies a single neural network to the full image. This network divides the given image into several regions and predicts bounding boxes in each region and probabilities for each region. These bounding boxes are weighted by the predicted probabilities. High scoring regions of the image are considered detections. YOLO is very accurate and fast with 45 Frames per second (FPS), with better performance than RCNN and other object detection algorithms.

For text localization and script identification, we consider Kannada, Hindi, and English as 3 different classes, and the dataset is constructed by considering images of the above-mentioned languages with different text orientations, image

backgrounds, and image resolutions. The model is first trained with images having English, Kannada, and Hindi characters and then with the words. The training dataset consists of 1149 images with more than 3000 labels. The trained model is then tested on street-view videos to check the performance. The experimental results are discussed at the end of the paper.

APPLICATIONS

Over the years, there have been many text-related applications for both color images and videos. Some of the applications of text detection and its script identification are addressed below.

Translation for Tourists: Many countries are visited by tourists to understand their culture, to enjoy and spend vacations, etc. It is also important for the countries to bring in tourists as a means to increase revenue. Tourists often find it very hard to find the destination and also read the billboards and name boards of shop in foreign places as they are not familiar with the language. India is the resident of many regional languages like Tamil, Telegu, Bihari, Gujarati, Kannada, Marathi etc. along with national language – Hindi. As a part of translation, the text in the scene images has to be detected first and the script needs to be recognized to translate it to user-choice language which would provide better knowledge and experience to the tourists and would also help them to have better awareness of their surroundings.

Guide for visually impaired: Often visually impaired people do not like to depend on others for their day-to-day chores. They need electronic guidance that would guide them in shops, markets, and detect their surroundings. It's often very hard for the visually impaired to know the item labels and shop names, this can be achieved with computer vision where the item labels and shop names would be detected and translated with voice-over for them to understand and have a comfortable experience.

Multimedia retrieval [2]: Captions in videos, webpages and images provide important information about the multimedia content they also annotate the information about where, when, and who of the happening events. Recognizing the text and extracting them from such multimedia sources would better enhance multimedia retrieval.

Automation in Industries [2]: Recognizing text and identifying the scripts on the packages, containers, grocery items in marts, and maps has wide applications in industrial automation. For instance, the envelopes having addresses in different languages can be extracted and converted to different languages for better mail sorting systems. To improve the logistics efficacy, the identification of container numbers can be really helpful. Identification of house numbers and text in maps can be benevolent in the area of automatic geocoding systems.

CHALLENGES

Text detection and script identification in complex natural scenes can be very challenging. The text localization in color images has many complexities like different background colors in image, scene complexity, different fonts of text, different text orientations, light glares and lens flares in images, illumination effect in images etc. Some of the challenges found in text localization and script identification in color images and videos are:

Complexities	Environment	Image capture	Text content
	Scene intricacy	Lens flare and light glares in images	Different orientations
	Uneven illumination	Resolutions/burring	Fonts variations
		Image distortion	Multi-lingual texts

OBJECTIVES OF THE PROJECT

1. Construction of training dataset with images containing text in different languages like Kannada, Hindi, and English and subsequent labels.
2. Training the model to detect and classify text in a picture or video based on its language (Kannada, Hindi, and English).
3. Test the model on graphic text videos and pre-recorded real-time scene videos.

4. Test the model to detect text in a variety of situations, including blurred photos, pictures with varying backgrounds, text orientations, and font sizes and styles.
5. Evaluate the model with Tensorboard.
6. Comparison of proposed model with existing models.

LITERATURE SURVEY

[1] *Pushpalatha M and Dr. Antony Selvadoss Thanamani presented ‘Geometrical Features for Detection of Kannada Text in Images and Documents’* which discusses the Geometrical Machine Learning Algorithm for Kannada text detection. It analyzes and understands the Kannada Text and analyzes the geometry of the Kannada Characters to decide the location of the Kannada Text, which matches the structure of the trained geometry in the form of supervised machine learning. The proposed algorithm secures an accuracy score of 74.9 percent and is compared with other methods like K-means, Gibbsian Extraction Model.

[2] *Qixiang Ye and David Doermann presented ‘Text Detection and Recognition in Imagery: A Survey’* which addresses the technical challenges faced in text detection and recognition in color imagery. It analyses and compares existing techniques’ performances and also categorizes them as either step-wise or integrated methodologies. Text categories and subcategories are shown, benchmark datasets are listed, and the performance of the most representative approaches is compared.

[3] *M.C. Padma and P. A. Vijaya proposed ‘Script Identification of Text Words from a Tri-Lingual Document Using Voting Technique’*. In this paper, a proposed model is used to identify and separate text words of Kannada, Hindi, and English scripts from a printed tri-lingual document. The presented method is trained to thoroughly learn the well-defined characteristics of each script and uses a simple voting method for classification. The results of the proposed method are found to be 99% on a manually created dataset and 98.5% for the dataset constructed from scanned document images.

[4] *Khanaghavalle. G. R and Dr. N. Rajeswari presented ‘Arbitrary Shape Hindi Text Detection for Scene Images’* by proposing a novel Hindi Text Detector using ResNet as the backbone network. The presented method is able to locate arbitrary shaped Hindi text in complex background images and has been experimented with the benchmark dataset IC19-MLT (Hindi). This method is able to achieve high accuracy from 74% to 78.5%.

[5] *Huibai Wang and Hongqing Shi presented ‘Research on text detection method based on improved yolov3’*. In this paper a method is proposed for text detection in natural scenes based on UDSP- YOLO, along with this it uses a CLAHE image enhancement preprocessing method to remove the effect of lighting changes in images and also uses S3Pool for feature extraction. The result of the proposed method is proven to have 0.653 Precision and 0.451 Recall with 0.534 F-measure.

[6] *Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman presented ‘Synthetic Data for Text Localisation in Natural Images’* by introducing a fully convolutional regression network (FCRN) for text recognition and bounding box regression at all positions in an image and at multiple scales. The proposed method is compared with other end-to-end object detection systems. FCRN achieves an F-measure of 84.2% on the standard ICDAR 2013 benchmark dataset.

[7] *Tianxiang Zhou, Ke Wang, Jun Wu, and Ruifeng Li proposed ‘Video Text Processing Method Based on Image Stitching’* by introducing a method for obtaining video text panoramas which then processes video text content by natural scene text tracking, text positioning, image stitching, and other methods. Text detection is accomplished by using the YOLO target detection framework and the ECO tracking method is used to keep track of texts. With the global stitching method, the text panorama is achieved. However, the image stitching needs to be improved as it does not work efficiently for the blurring of the frame caused by the video movement.

[8] *B.V.Dhandra, H.Mallikarjun, Ravindra Hegadi, and V.S.Malemath presented ‘Word-wise Script Identification from Bilingual Documents Based on Morphological Reconstruction’* by proposing a script identification at the word level based on morphological reconstruction for 2 bilingual documents containing Kannada and English scripts. The proposed technique includes a feature extractor and a classifier. The k-nearest neighbor algorithm is used to classify the new word images. The proposed algorithm is checked on 2250 sample words with different font styles and sizes. The results show that the accuracy of Kannada text recognition is 90.3% while for English and Hindi is 96.9% and 95.6% respectively.

[9] *Archana Shirke, Paresh Pandit, Nikunj Gaonkar, and Kapil Parab proposed ‘Handwritten Gujarati Script Recognition’* which uses the YOLO algorithm for the recognition of Handwritten Gujarati script from images and documents. This involves creating a neural network that takes the input as an image and extracts features from it. The neural network recognizes this text and outputs it to the machine. As the YOLO darknet framework is used, the accuracy is improved, speed is increased and time is reduced.

METHODOLOGY

A. DATA COLLECTION

There are many videos and natural scenes available having the tri-lingual scripts i.e. English, Hindi, and Kannada. But the standard database having the images of these tri-lingual scripts is scarcely found. For the proposed model, two sets of the database were constructed where one was used for training the model while the other database was used for validation. The images were collected from several sources. Most of the images with horizontally oriented scripts were acquired from translation videos and the natural scene images were obtained from the videos having street-view data of places located in Karnataka, Varanasi, Delhi, and Mumbai. The natural scene images have scripts of different orientations, colors, backgrounds, and also with varieties of image resolutions. The images with noise and disturbances were also

used to train the model to make it robust and efficient. The training database has images having characters of Hindi, English, and Kannada as well as words from these scripts for better script identification purposes.



Fig1. Images from translation videos



Fig2. Images from recorded street videos

There are two essential parts desired for training the model and its testing i.e. the images having tri-lingual scripts and corresponding labels for each text in an image. To train the models, we need labels of each text in an image that describes the location of that text in the image. These labels are formatted in such a way that the model would automatically be able to locate the objects/scripts in the image and draw the ground truth bounding boxes while training. The collected images have varying sets of texts. Each image is at least associated with 3 labels and has the utmost 40 labels. The model is trained with characters as well as words from all 3 scripts. Some of the images used in the dataset are displayed in the above figures.

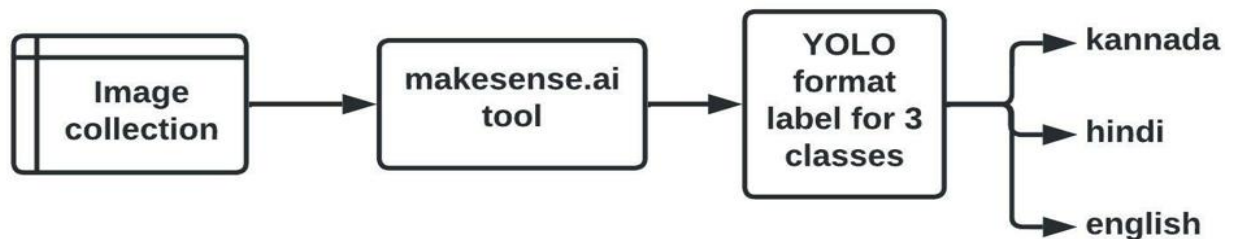


Fig3. Overview of Data Collection

Each image in the dataset has a corresponding label text file (*.txt) in YOLO format. The annotations are created using an external tool called ‘makesense.ai’. In this tool, different objects/text scripts are considered as different classes. We have labeled the texts found in an image as ‘Hindi’, ‘English’, and ‘Kannada’ classes. The YOLO label comprises 5 parameters (x, y, w, h, c) where, (x, y) are the upper-left corner coordinates of the bounding box, (w, h) are the width and height of the bounding box having the text and c corresponds to the object class. The training dataset has 1157 images and 1157 text files containing labels. Out of which 1139 images are used for training the model and 18 images are used for testing the trained model. The structure of the entire dataset is shown below.

DATASET	IMAGES		LABEL TEXT FILES	
	Test	Validation	Test	Validation
	1139	18	1139	18

The dataset is constructed in a way to make the model more robust and competent to localize the texts in an image and identify their scripts accurately. The total size of the dataset is 1.803 GB. This dataset along with a custom YAML file that consists of information like the directories path where the data set is located, the number of classes and the name of each class are fed as an input to the model.

B. YOLO MODEL ARCHITECTURE

Yolo is a single-shot object detection algorithm as it processes an entire image once using a single CNN. YOLO is very fast in computation and can be used in a real-time environment. This algorithm has a very high accuracy range and has good performance with 45FPS. Basically, YOLO works by dividing the input image into grids of equal size ($S \times S$) and then it predicts a class and a bounding box of objects present in the grid for every grid in the image. The input to the YOLO is images with corresponding labels (ground truth bounding boxes) and then YOLO will predict the bounding boxes with confidence scores and a class probability map for each image. Finally, the output obtained from YOLO on the test image contain final detections with bounding boxes around the detected objects in an image.

[5] YOLO converts the target detection problem into a logistic regression problem, to accomplish a deep convolution network and to achieve fast object detection which ensures higher overall accuracy. YOLO scales the original image into a size of 416x416. The feature extraction network divides this image into $S \times S$ grid cells having equal sizes according to the scale of the feature map. The feature map size of scales can be 13x13, 26x26, and 52x52. Then the union of shallow and deep features is used to achieve more discriminative deep features. Lastly, each cell draws 3 anchor boxes to predict 3 borders as a part of regression prediction.

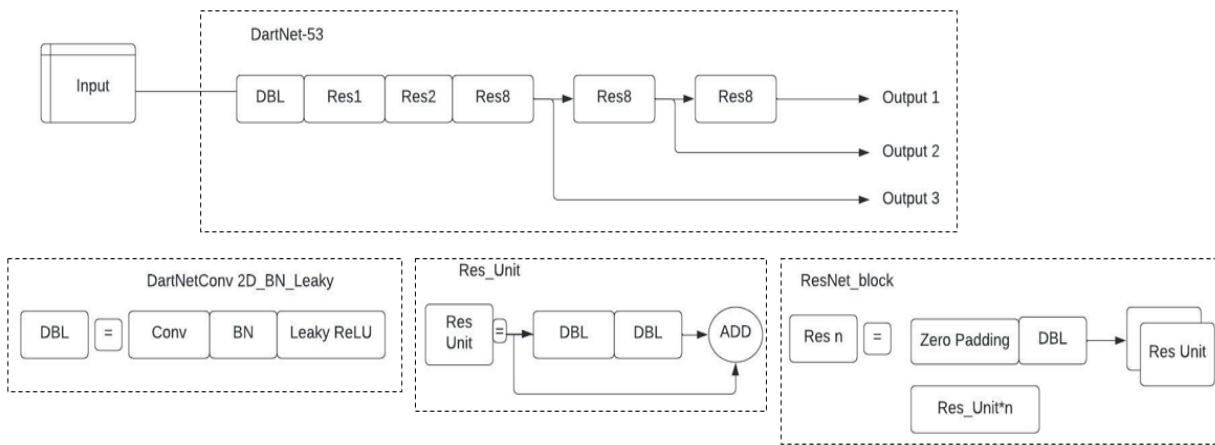


Fig 4. YOLO Darknet-53 architecture

For each anchor box, Yolo predicts 4 values that are logged as (tx, ty, tw, th). These values are coordinates of the upper left corner of the border (tx, ty) and the width and height of the target (tw, th). [5] If the target center/object center in the cell is offset from the upper left corner of the image (Cx, Cy) and the anchor box has width and height (pw, ph), then the modified borders are:

$$bx = \sigma(tx) + Cx$$

$$by = \sigma(ty) + Cy$$

$$bw = p_w \cdot e^{tw}$$

$$bh = p_{h.e}th$$

The selection of the anchor box is done by the method of dimension clustering and the traditional clustering algorithm which includes K-means clustering, hierarchical clustering, and model-based method. In this algorithm, K-means clustering is used to cluster the size of the object/target border in the training set to attain the finest anchor frame size, in order to predict more target borders. The distance measure of K-means clustering is as follows

$$d(\text{box}, \text{centroid}) = 1 - \text{IOU}(\text{box}, \text{centroid})$$

Here, IOU refers to Intersection over Union while the box is a border size sample in the dataset and the centroid denotes cluster center size.

So as to predict the probability of an object in the anchor box, Yolo uses logistic regression. If the rate of overlap of the anchor point frame and the real object frame is greater than any other anchor point frame, the probability of this anchor point frame is 1. If the overlap rate of the real target/object frame and the anchor point frame is greater than 0.5 then this prediction is ignored. This algorithm uses binary cross-entropy loss and logistic regression to achieve category prediction during the training process. This allows YOLO to achieve the multi-label classification of a target/object.

C. PROPOSED MODEL

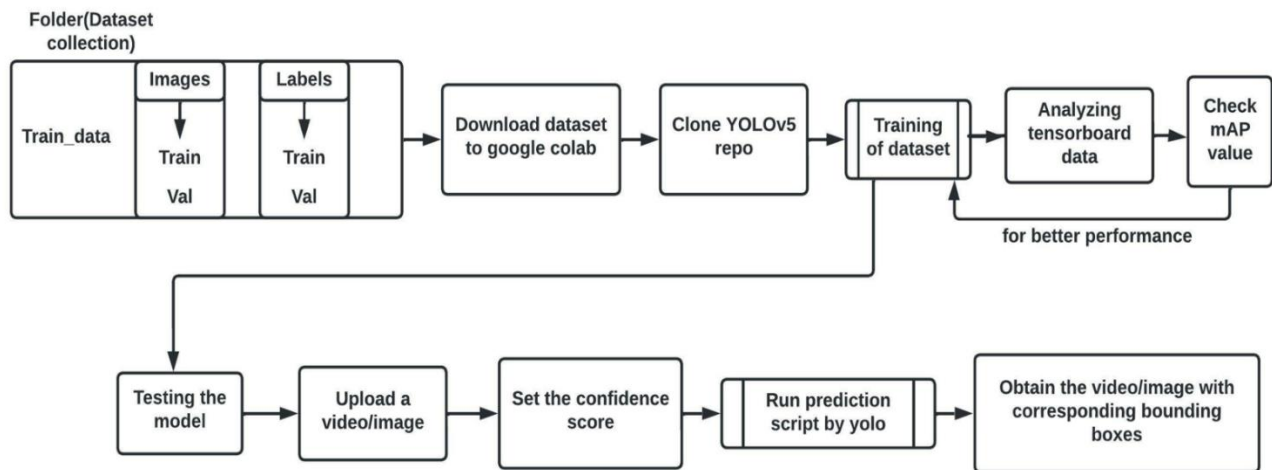


Fig 5. Overview of the proposed model

The parameters set to train the model are:

No of GPUs: 01

No. of Classes: 3

Name of Classes: 'Kannada', 'Hindi', 'English'

Batch size: 16

Epochs: 100

Weights: yolov5s.pt

After the successful construction of the dataset, this training dataset is given to the model to train it. It is seen that the more the images and labels are given to the model, the more precise it is able to predict and identify the scripts. The environment used to run the model is Google Colab using 1 GPU as the hardware accelerator. Packages like Torch OS and ipython are imported to display the images. The darknet53 is installed which is the backbone of YOLOv5 and constitutes of Convolution Neural Network with 125 layers. The other requirements are installed into google collab which creates yolov5 file directory.

A custom file in YAML format is created with information such as class labels and a number of classes and is uploaded into the data file in the yolov5 file present in the repository. The model is trained after specifying image dimensions which are set to 416, the batch size is 16 and the number of epochs is 100. The location of the actual dataset and YAML file is specified. Here, yolov5s weights, which is a small model, are chosen and the cache is specified to cache the images in the GPU. Weights and biases are downloaded and the Yolo model starts training the convolutional neural network.

After setting the necessary parameters and the training of the model is initiated, each epoch run gives information like GPU memory, classes, number of labels, image size, and mAP values with a confidence score of a bounding box that is predicting the text in the images. After the epoch runs are completed, the last and the best weights are obtained. By validating the best weights, we get the model summary. Results of training are stored in a trained file in the repository. If the model does not give satisfactory results, batch size and number of epochs are changed and the model

is trained again. The results are visualized with the tensorboard to get metrics and graphs which helps us to know if the model still needs to be trained.

Testing of the model is done by uploading the sample video or images into the colab repository. The detection script is set to run by specifying the sample video or images by choosing the best weights from the training process of 100 epochs. The confidence score is set over 0.2 for the correct prediction. The result obtained from the prediction script is stored in the detect file in the repository. The sample video/image is downloaded from the repository which contains the text which is localized and identified using the bounding box. The results obtained from the model are discussed in the below section.

RESULTS

A. EXPERIMENTAL RESULTS

To test the efficacy of the model, the trained model is tested on several graphic text translation videos and real-time recorded street view videos. The model has a speed of 45 frames per second and hence it is able to detect texts in the video very fast and also identifies the script of the localized text. The screenshots of the sample video and the Yolo detections on the video are displayed. The YOLO detects the text by drawing the bounding box around it and identifies the script as one among 3 classes and displays the confidence score for each detection. The first video that was used for testing was English, Hindi, and Kannada translation of sentences. Before and after detection are shown below.

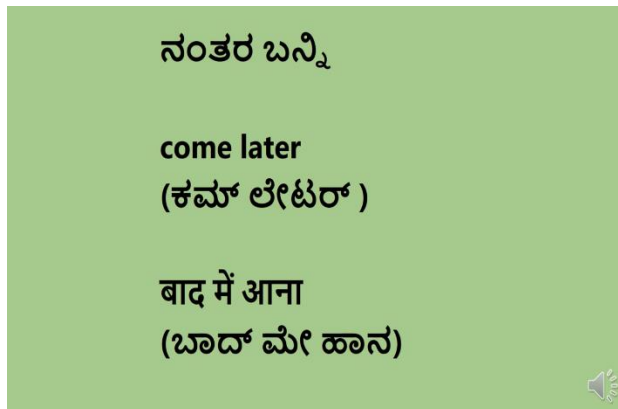


Fig 6. frame screenshot of sample video



Fig 7. Detected frame of sample video

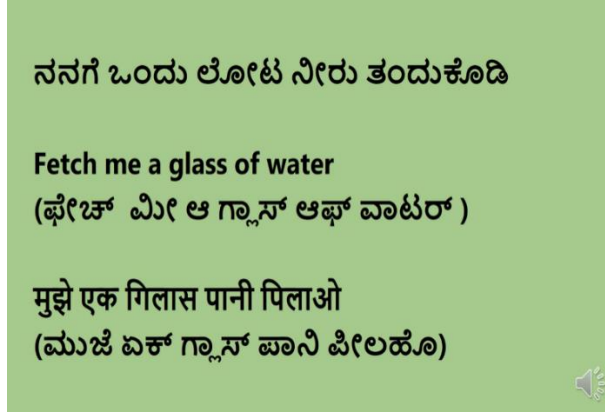


Fig 8. frame screenshot in sample video



Fig 9. Detected frame in sample video

The second video that was used for testing was English, Hindi, and Kannada translation of words with images of vegetables. The trained model has to treat the images of other untrained objects as background and only detect the texts. Before and after detection are shown below.



Fig 10. frame screenshot of sample video



Fig 11. detected frame in sample video



Fig 12. frame screenshot in sample video



Fig 13. detected frame in sample video

Lastly, the trained model was also tested on real-time street view videos of places like Varanasi, Mumbai, Delhi, and Karnataka, India. The video consists of the text of scripts that are in different orientations and complex backgrounds. Furthermore, the model was also checked on night and day street videos to check its detection accuracy. The results are displayed below.



Fig 14. frame in Varanasi street video

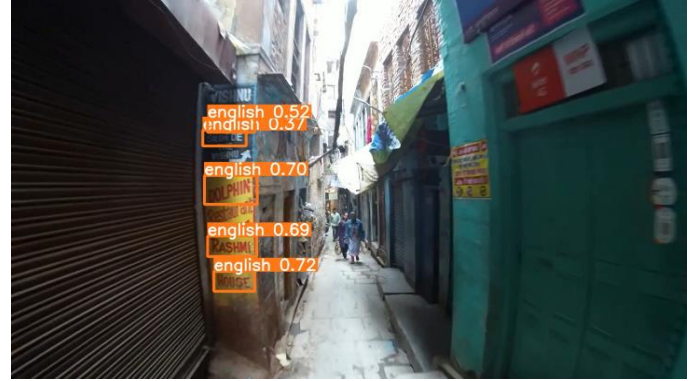


Fig 15. detected frame in Varanasi street video



Fig 16. frame in daytime Karnataka street video



Fig 17. Detected frame in Karnataka street video

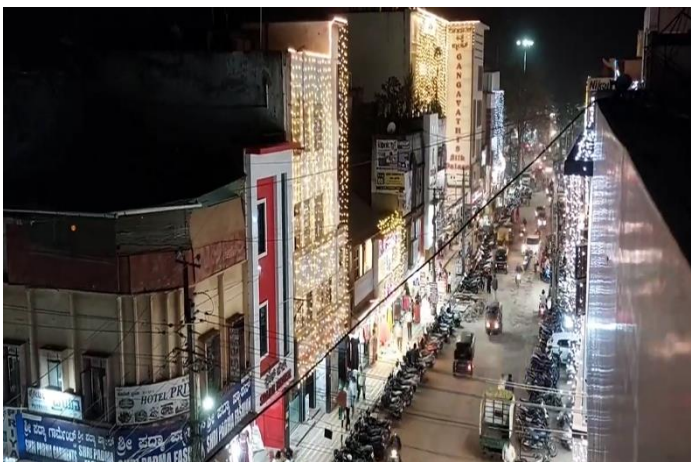


Fig 18. frame in night-time Karnataka street video



Fig 19. Detected frame in Karnataka street video

To check the robustness of the trained YOLOv5 model for text detection and script identification, the model was tested on different scenarios such as different color backgrounds, different font texts, different orientations of the text, disturbances (different lightings in images) in the images and different resolutions, etc.

The results are shown below, the undetected sample images are shown in the left side while the detected images by Yolo are shown on the right side.

(1) Different color backgrounds



Fig 20. sample image in beige background



Fig 21. detected sample image

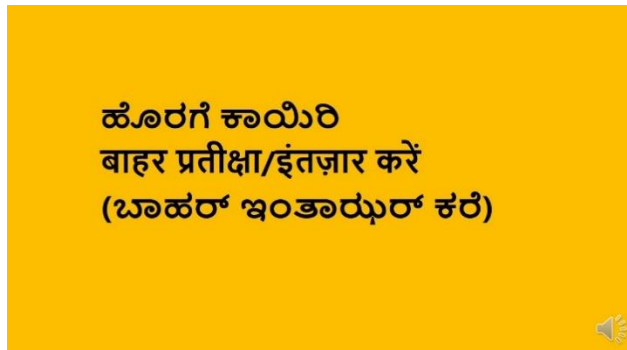


Fig 22. sample image in yellow background



Fig 23. detected sample image

(2) Different Fonts



Fig 24. different font in sample image



Fig 25. detected sample image



Fig 26. different font in sample image



Fig 27. detected sample image

(3) Light glare and illumination effect in images



Fig 28. light glare in sample image



Fig 29. detected sample image



Fig 30. illumination effect on sample image



Fig 31. detected sample image

(4) Different Resolutions

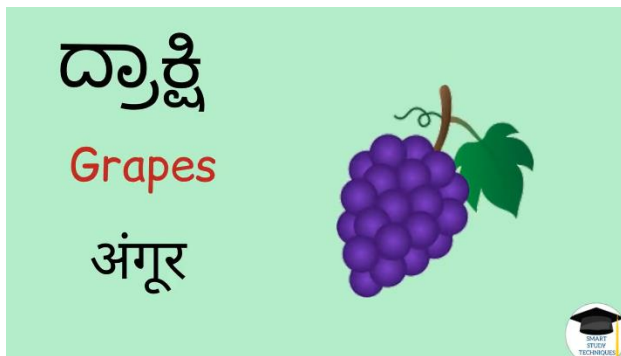


Fig 32. High resolution sample image

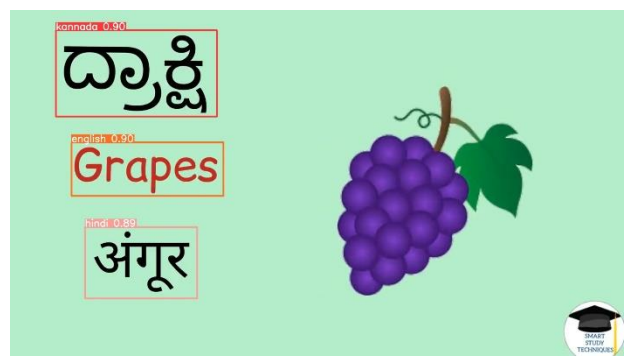


Fig 33. detected sample image



Fig 34. Low resolution (30% blur) sample image

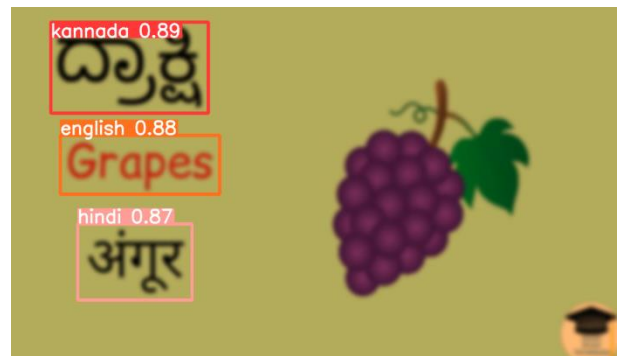


Fig 35. detected sample image

(5) Text orientations



Fig 36. sample image with titled horizontal text



Fig 37. detected sample image



Fig 38. sample image with vertical text orientation



Fig 39. detected sample image

B. TENSORBOARD RESULTS

Each object has a bounding box and a corresponding class label. Object detection systems make predictions in the footings of a bounding box and a class label. After the training of the dataset, the results and predictions are visualized using Tensorboard. TensorBoard is TensorFlow's visualization toolkit, which enables us to track metrics like loss and accuracy, envision the model graph, view histograms of weights, biases, or other tensors as they change over time, and much more. It is an open-source tool that is part of the TensorFlow ecosystem. TensorBoard is used to:

- Recognize the flow of tensors
- Debugging
- Optimizing

Using tensorboard, the model which gives better accuracy in the results can be determined. Also, the mAP of the trained dataset can be realized.

mAP (Mean Average Precision):

In computer vision, mAP is a prevalent evaluation metric used for object detection (i.e. for localization and classification). **Localization** finds the location of an instance (e.g. bounding box coordinates) and **classification** tells what the object is by labeling it. To determine the mAP of a model, IoU is calculated

.

IoU (Intersection Over Union):

For each bounding box, the overlay among the predicted bounding box/anchor box and the ground truth bounding box is measured and this is achieved by IoU. In IoU, the threshold is to be established. In some datasets, we predefine an IoU threshold (say 0.5) in classifying whether the prediction is a true positive or a false positive.

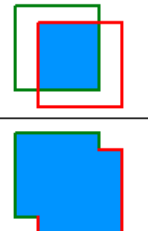
$$IOU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\text{area of intersection}}{\text{area of union}}$$


Fig 40. Intersection over union

- If $\text{IoU} \geq 0.5$, classify the object detection as True Positive (TP);
- If $\text{IoU} < 0.5$, then it is a wrong detection and classifies it as False Positive (FP);
- When ground truth is present in the image, and the model fails to detect the object, then it is classified as False Negative (FN);
- True Negative (TN): TN is every part of the image where we did not predict an object. As this metric is not helpful for object detection, we ignore it.

For instance, if IoU is 0.1, then the predicted bounding box is 10% away from overlapping with the ground truth bounding box. As the predictions are based on the bounding box and class labels. If the threshold is 0.5, and the IoU value for a prediction is 0.7, then we classify the prediction as a True Positive (TF). On the other hand, if IoU is 0.3, we classify it as a False Positive (FP).

Precision and Recall:

For prediction/ object detection, two values are required they are **precision** and **recall**. **Precision** measures how precise are the predictions. i.e., the percentage of predictions is correct. **Recall** measures how well all the positives are found. It is the ability of a model to find all the relevant cases within a dataset. The higher the precision, the more confident the model is when it classifies a sample as *Positive*. The higher the recall, the more positive samples the model correctly classified as *Positive*.

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{TP}{\text{all detections}}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

When a model has high recall but low precision, then the model classifies most of the positive samples correctly but it has many false positives (i.e. classifies many *Negative* samples as *Positive*). When a model has high precision but low recall, then the model is accurate when it classifies a sample as *Positive* but it may classify only some of the positive samples. A **Precision-Recall curve** shows the trade-off between the precision and recall values for different thresholds. This curve helps to select the best threshold to maximize both metrics.

Confidence score: The confidence score reflects how likely the box contains an object (objectness) and how accurate is the bounding box drawn around the object. If no object exists in that cell, the confidence score will be zero. The below images show the results obtained after training the dataset with bounding boxes and the class labels. The value beside the labels (Kannada, Hindi, English) represents the confidence score. The confidence score is between 0 and 1. A score nearer to 1 represents good accuracy of the bounding box.



Fig 41. image with confidence score on translation videos



Fig 42. Image with confidence score on recorded street videos

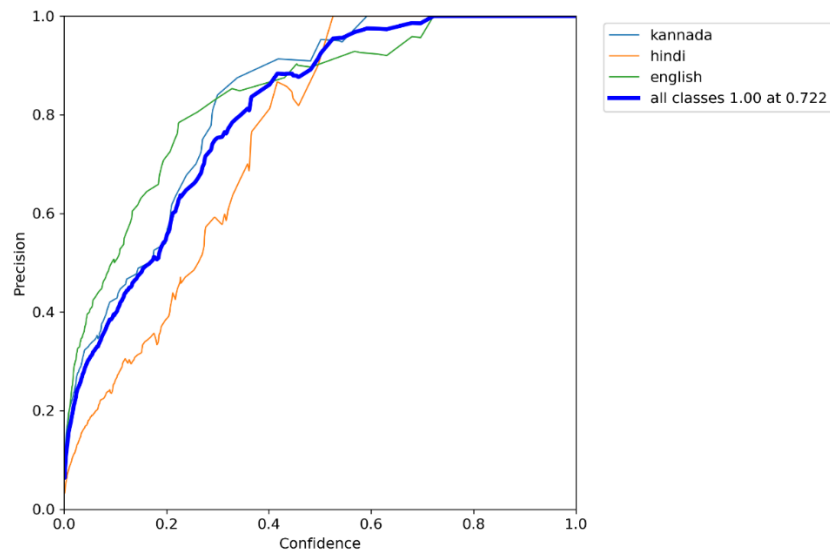
Graphs:**1. Precision-confidence curve:**

Fig 43. P_curve

Fig 1 represents the precision-confidence curve, it is seen that as the confidence score increases between 0 to 1, the precision increases i.e., true positive increases. The precision curve is obtained for each class label and also for the overall model.

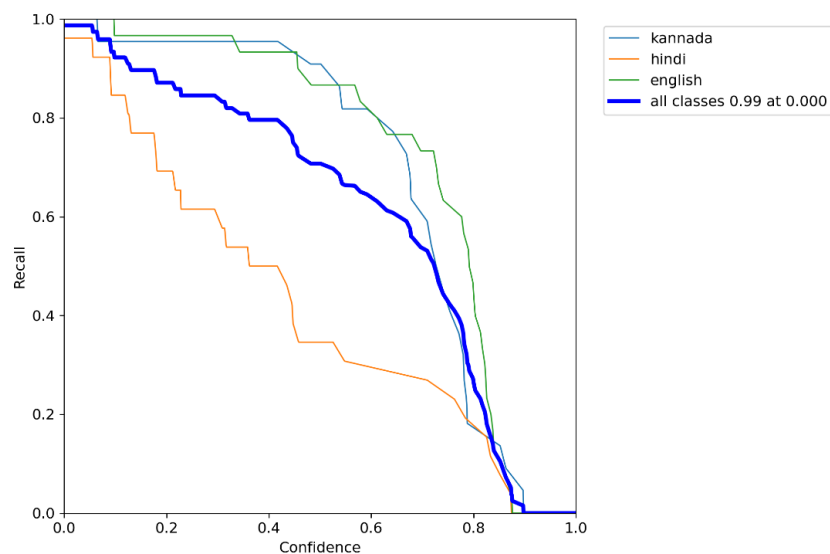
2. Recall-confidence curve

Fig 44. R_curve

Fig 2 represents the recall-confidence curve, it is seen that as the confidence score increases between 0 to 1, the recall decreases. The recall curve is obtained for each class label and also for the overall model.

3. Precision-Recall (PR) curve

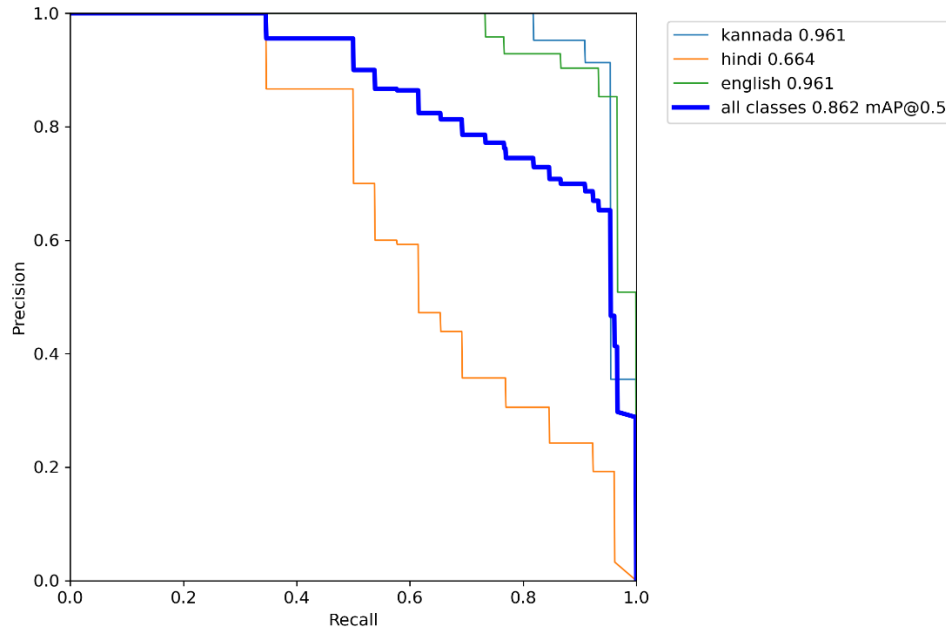


Fig 45. PR curve

A PR curve is a graph with precision values on the y-axis and recall values on the x-axis. It is desired that the algorithm should have both high precision, and high recall. However, most of the cases often involve a trade-off between the two. A good PR curve has a greater AUC (area under the curve). Fig 3 represents that the PR curve of all classes has a greater AUC with a value nearer to 1 at mAP 0.5.

mAP results:

the mAP is used to evaluate the performance of both classification and localization using bounding boxes with the help of the concept of IoU (Intersection over union). Here the IoU is 0.5 i.e. the threshold value

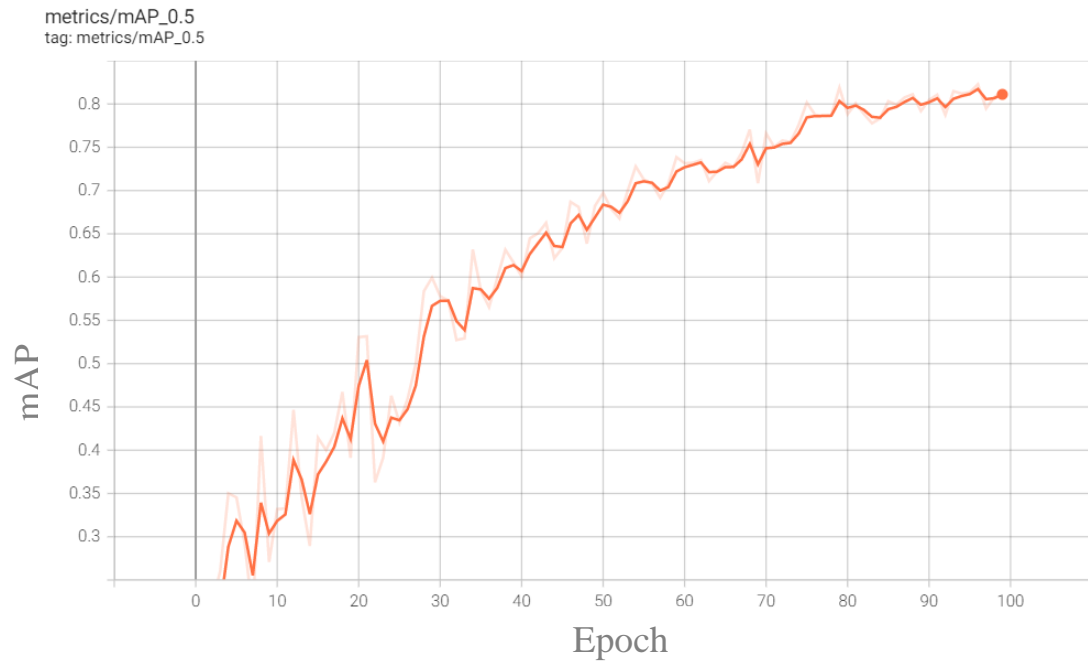


Fig 16. mAP_0.5 results

From Fig 4, the IoU of 0.5 mAP_0.5 or mAP@50 is calculated. As the epoch increases from 0 to 99, the mAP value gradually increases and reaches up to 0.811 which shows that the model prediction has a correctness of 81.1%.

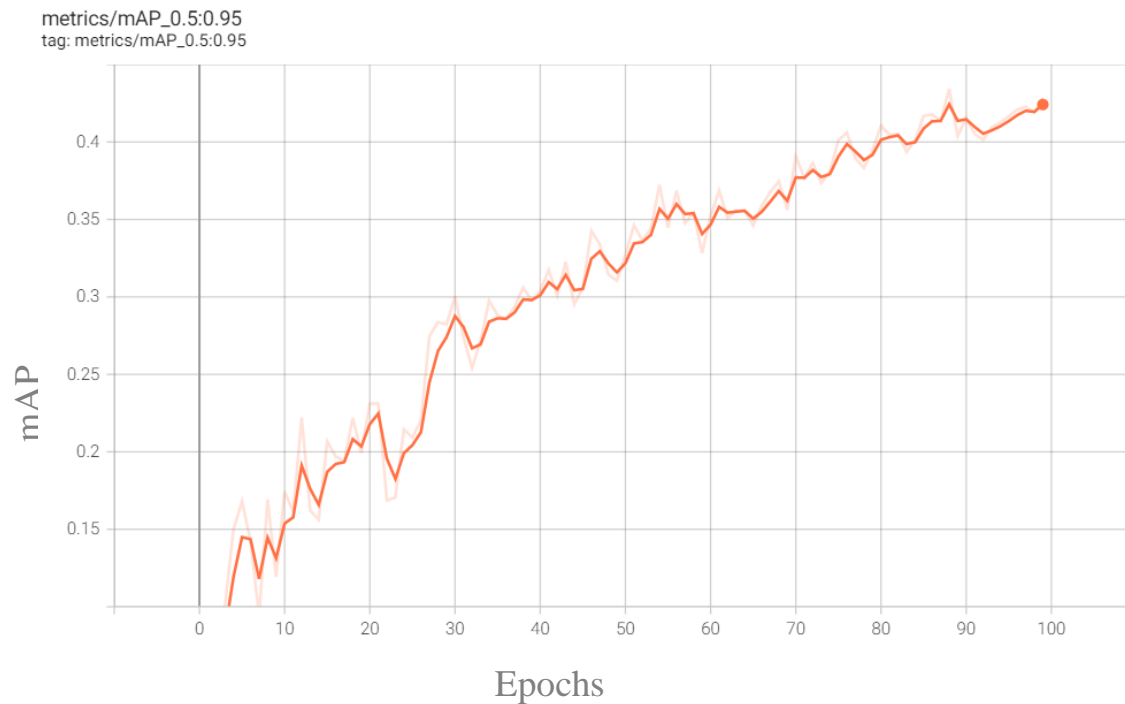


Fig 47. mAP_0.5:0.95 results

mAP@[.5:.95] means average mAP over different IoU thresholds, from 0.5 to 0.95, step 0.05 (0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95).

From Fig 5, it can be observed that for 100 epochs the value of mAP is 0.434.

Model summary: 213 layers, 7018216 parameters, 0 gradients, 15.8 GFLOPs

Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95:
all	18	78	0.784	0.75	0.811	0.434
kannada	18	22	0.817	0.909	0.889	0.487
hindi	18	26	0.667	0.461	0.58	0.249
english	18	30	0.868	0.878	0.965	0.565

The highlighted row represents the results of the overall trained dataset. The mAP is nearly equal to 1 ([mAP@.5](#) = 0.811 for all), which denotes that the trained model has good accuracy and it yields good results.

Train losses:

There are three different types of loss.

1. Box loss: The box loss represents how well the algorithm can locate the centre of an object and how well the predicted bounding box covers an object. The bounding box drawn on the text in this model is accurate, which makes the box loss approximately 0.

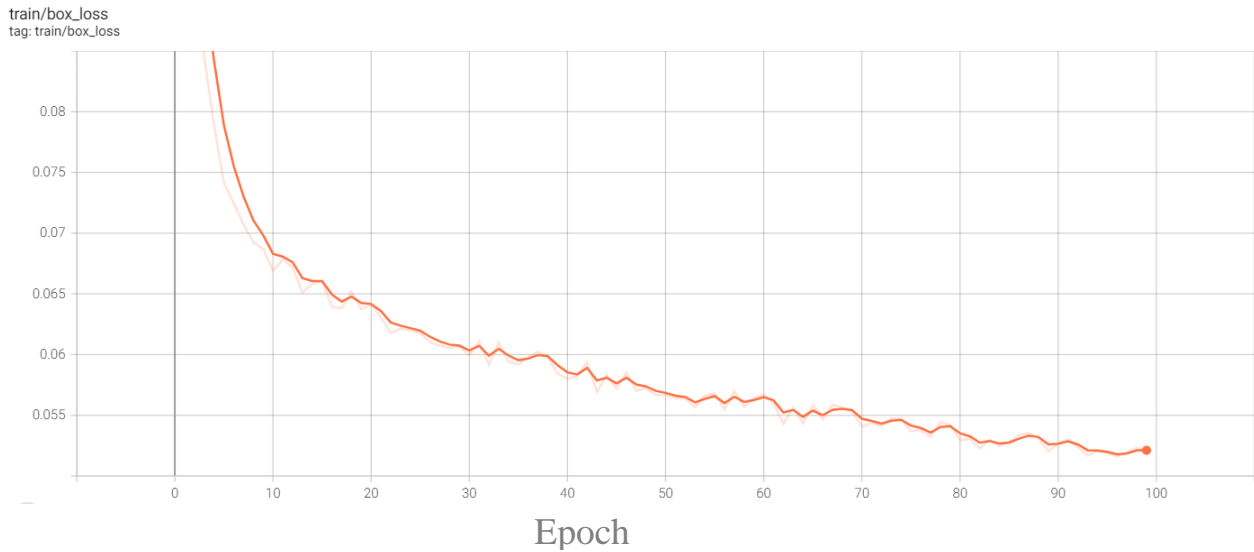


Fig 48. Box loss

2. Objectness loss: Objectness is essentially a measure of the probability that an object exists in a proposed region of interest. If the objectivity is high, this means that the image window is likely to contain an object. This model can recognize the object and classify it accordingly which gives nearly 0 objectness loss.



Fig 49. Objectness loss

3. Classification loss: Classification loss gives an idea of how well the algorithm can predict the exact class of a given object. The model can classify the kannada, hindi and english classes, hence the classification loss is nearly 0.

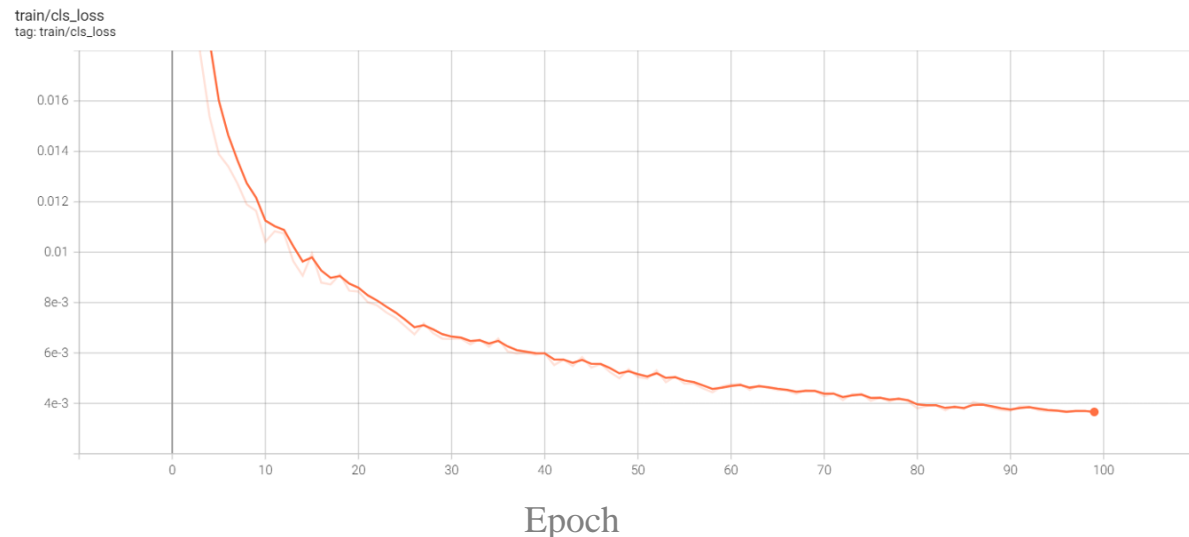


Fig 50. Classification loss

C. COMPARISON WITH OTHER METHODS

The proposed model was compared with other existing text detection methods. While there are not many models proposed which detects the text of tri-lingual scripts in natural scene images and in videos. Hence few proposed models for single script identification by the previous researchers are included for comparison.

SI No.	Methods	Accuracy – Kannada	Accuracy – Hindi	Accuracy – English
1.	Geometry based Kannada Text Detection [1]	74.9%	-	-
2.	Region Proposal Network (RPN) & feature pyramid network using ResNet [4]	-	74%	-
3.	UDSP-YOLO [5]	-	-	95%
4.	Proposed YOLOv5 model	88.9%	58%	96.5%

The **overall accuracy** of the proposed model for text detection and script identification is found to be **88.1%**.

CONCLUSION

The research work has shown substantial contribution in text localization and tri-lingual script identification in natural scene images as well as videos. Despite the complexities in the images and intricacies in the curves of Kannada, Hindi, and English characters the proposed model is able to detect the text and identify the script successfully. Some of the most challenging issues in text detection of natural scene images have been addressed, including different backgrounds of the image, text orientations, different fonts of text, different resolutions, and disturbances in the image that makes the text detection in the image more complex. The proposed model provides accuracy of 88.1%-95% based on increase in dataset and tweaking few parameters. However, there is still room for improvement for better script identification of these tri-lingual scripts. The text detection in real-time images and video has wide variety of applications like text extraction and translation to

subsequent user-choice languages which would help tourists and the visually impaired to have a better understanding of the surroundings.

FUTURE SCOPE

Even though the proposed model is able to detect tri-lingual scripted texts in color images and videos and achieves an accuracy of 88.1%-95%. There is still wider scope to improve the model for text detection and script recognition.

Increase the size of the training dataset: We have included around 1150 images to train the model, detect the text and classify the text as either English, Hindi, or Kannada which yielded the above-mentioned accuracy. However, the accuracy can still be improved by increasing the size of the training dataset, including more natural scene images that tackle all the complexities. This would make the model more robust and its functionality can be greatly increased.

Different styles of texts: There are varieties of fonts and stylings associated with the text. The proposed model is trained for a limited number of fonts and styles of text found in color images. There is still scope for a model that is able to recognize text having different fonts and styles with different backgrounds in natural scene images and other color images as well as real-time videos.

Detect all Indian languages: As we have trained the model to detect tri-lingual script texts like Hindi, Kannada, and English, there is a need for a model that is able to detect all Indian languages as India houses many regional languages and these are used widely all over India in billboards, shop labels, etc. There is room for a better model that would be able to detect all the Indian languages.

Cursive characters detections: The proposed model is able to detect text having few cursive characters. As there are many variations in cursive characters for all languages. There is a scope for the model that is able to detect the normal text as well as the cursive text so that it is able to detect even hand-written scripts.

Similar language detection: Regional Indian languages like Kannada and Telugu has similar curves in their characters which would make it hard for the trained model to accurately identify the script. There is a need for a more robust and efficient model that would accurately classify the text than giving out incorrect predictions.

REFERENCES

- [1] Pushpalatha M, Dr. Antony Selvadoss Thanamani. Geometrical Features for Detection of Kannada Text in Images and Documents. In International Journal of Scientific Research and Review, Volume 7, Issue 11, 2018.
- [2] Qixiang Ye, David Doermann. Text Detection and Recognition in Imagery: A Survey. In the proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence 37, 2015.
- [3] M.C. Padma, P. A. Vijaya. Script Identification of Text Words from a Tri-Lingual Document Using Voting Technique. In International Journal of Image Processing, March 2010.
- [4] Khanaghavalle. G. R, Dr. N. Rajeswari. Arbitrary Shape Hindi Text Detection for Scene Images. Published in International Research Journal of Engineering and Technology (IRJET). Volume: 07, Issue: 05, May 2020.
- [5] Hubai Wang, Hongqing Shi. Research on text detection method based on improved yolov3. In Proceedings of IEEE 5TH Advanced Information Technology, Electronic and Automation Control Conference, 2021.
- [6] Ankush Gupta, Andrea Vedaldi, Andrew Zisserman. Synthetic Data for Text Localisation in Natural Images. In 2016 IEEE Conference on Computer Vision and Pattern Recognition, 2016.
- [7] Tianxiang Zhou, Ke Wang, Jun Wu, Ruifeng Li. Video Text Processing Method Based on Image Stitching. In proceedings of IEEE 4th International Conference on Image, Vision and Computing, 2019.
- [8] B.V.Dhandra, H.Mallikarjun, Ravindra Hegadi, V.S.Malemath. Word-wise Script Identification from Bilingual Documents Based on Morphological

Reconstruction. In Proceedings of 1st International Conference on Digital Information Management, 2006.

[9] Archana Shirke, Nikunj Gaonkar, Paresh Pandit, Kapil Parab. Handwritten Gujarati Script Recognition using YOLO. In 7th International Conference on Advanced Computing and Communication Systems, 2021.

[10] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.