Journal of Innovative Image Processing (ISSN: 2582-4252)
www.irojournals.com/iroiip/

A Deep Learning Framework for Kannada-English Text Recognition and Language Identification in Natural Scene Images

Venkata B Hangarage[1], Gururaj Mukarambi[2*]

Department of Computer Science, School of Computer Science, Central University of Karnataka, Kadaganchi, Aland Road, Kalaburagi, Karnataka, India.

E-mail: [1]vbj44@yahoo.com, [*2]gmukarambi@gmail.com

Abstract

Natural Scene Text Detection and Language Identification is a challenging problem in the field of computer vision, due to autonomous video surveillance and the design of an OCR system for natural scene images. The drawback of an autonomous video surveillance and monolingual OCR system is that it will not work efficiently on natural scene images, where text appears in different orientations, backgrounds, and lighting conditions with multilingual scripts. Hence, we proposed a deep learning model, i.e. fine-tuned YOLOv5, for text detection and language identification in bilingual scene images. For testing the proposed (fine-tuned) model, there is no standard ground truth database in the literature. Therefore, we created our own real-time natural scene dataset from the Kalaburagi and Bidar districts in the state of Karnataka. The proposed (fine-tuned) model involves training YOLOv5 on a real-time dataset, and it works with a genetic approach. It produces the anchor boxes for the objects present in the natural scene image. To test the performance of the fine-tuned YOLOv5 model, we employed evaluation metrics like precision, recall and accuracy. The experimental setup demonstrates robustness of the fine-tuned YOLOv5 model for text detection and language identification. We obtained an optimized precision rate of 86.8%, a recall rate of 83.4%, an F1 score of 85%, and an accuracy of 94.4%. The training of 80% and testing of 20% was carried out in the experiment. A comparative analysis of the fine-tuned YOLOv5 model with existing methods found in the literature is carried out, and observed that the fine-tuned YOLOv5 model shows better performance. The novelty of the paper is that the fine-tuned YOLOv5 model and dataset were constrained with a mixture of low-resolution and complex background images.

Keywords: YOLOv5,SPPF, Deep Learning, Computer Vision, Image Processing.

1. Introduction

Natural Scene Text Detection and Language Identification is an essential task in the field of computer vision, with profound implications for multilingual applications like automatic license plate detection, automatic street sign translation and assistance for visually impaired people. There is a pressing need to develop an autonomous system to detect and categorize the languages present in natural scene images to meet the requirements of multilingual text detection and language identification for the development of an OCR system. Hence, it motivates us to design and develop a proposed fine-tuned YOLOv5 for bilingual (Kannada and English) text detection and language identification in natural scene images.

*** Corresponding Author**

Journal of Innovative Image Processing, September 2025, Volume 7, Issue 3, Pages 976-990
DOI: https://doi.org/10.36548/jiip.2025.3.021
Received: 18.07.2025, received in revised form: 22.08.2025, accepted: 08.09.2025, published: 23.09.2025

The YOLOv5 model is an evolution of the YOLO architecture; it gains prominence due to its speed, efficiency, and adaptability in real-time object detection tasks. With its streamlined design process and enhanced performance, YOLOv5 presents an absorbing framework for the fusion of object detection and language identification in natural scene images.

The novelty lies in three aspects: (i). Customized anchor box scales and detection layers designed to capture small, elongated, and script-specific word shapes, (ii). A lightweight language identification head integrated within the detection network, enabling simultaneous localization and script classification without a separate OCR stage, and (iii). Robustness against script-specific challenges such as font variability, connected characters in Kannada, and overlapping bilingual words in noisy backgrounds. These contributions collectively enhance localization precision and script identification accuracy in real-world bilingual scenarios, advancing beyond conventional multilingual text detection frameworks that typically rely on post-processing or cascaded models.

There are four sections in the paper. Section 1 presents the introduction; data collection and its analysis are given in Section 2, the proposed methodology appears in Section 3, and experimental results and discussion appear in Section 4. Lastly, Section 5 provides the conclusion.

1.1 Background Study

Nisar K et al. [1] proposed a deep learning-based algorithm for Arabic and Pashto text detection. First, a dataset with a Pashto document image was created. After that, fine-tuning of convolutional neural network-based deep learning models like YOLOv5, YOLOv7, and SSD was performed. Further, they obtained text detection accuracies of 88.50%, 91.30%, and 84.5%, respectively.

Karan Maheshwari et al. [3] proposed a method using invariant moments. A two-step procedure is used in this study's technique to identify multilingual text in a natural scene image. First, a mixture of statistical filters is employed to select text patches from the image, improving the system's recall. To increase the system's accuracy, these areas are then run through an ANN classifier. The MSRA-TD500 dataset, containing both English and Chinese text, was used to test the system. An encouraging F1 score of 0.67 was returned when the algorithm's performance was assessed using the F1 score.

Joseph Raj et al. [4] proposed a bilingual text detection approach that employs Faster R-CNN to extract candidate text regions from natural scene images. These regions are rearranged as consecutive frames along the time axis, and both global and local shape features are captured using the Pyramid Histogram of Oriented Gradients (PHOG). A lightweight classifier is then applied to distinguish text from non-text regions. Performance evaluation on the MSRA-TD500 dataset, using the F1 score as the metric, demonstrates that the method achieves an F1 score of 0.70, showing improved accuracy compared to several existing text detection techniques.

Chandio, A. et al. [5] proposed bilingual Urdu-English text detection in natural scenes. It uses a CNN for feature extraction, followed by a bidirectional GRU with 512 hidden units and a Connectionist Temporal Classification decoder for text recognition. Language identification is implicit, relying on script-specific pre-processing to differentiate Urdu (cursive, right-to-left) and English (left-to-right). A custom Urdu-English dataset was created by augmenting a unilingual Urdu dataset (59,703 Urdu characters, 42,297 Urdu words) with

945 natural scene images containing 14,224 English characters, resulting in 2,835 images. Annotations include bounding boxes and transcriptions. Additional datasets include ICDAR2015 and COCO-Text for benchmarking. Character Recognition Rate (CRR): 98.5% for Urdu, 99.2% for English. Word Recognition Rate (WRR): 97.2% for Urdu. Outperforms baseline HOG-based methods (73.0% CRR) and unilingual CNNs (88.6% CRR).

Alshareef et al. [6] proposed bilingual Arabic-English text recognition in natural scenes, integrating detection, recognition, and language identification. It uses EfficientNetV2-L as the backbone for feature extraction, BiLSTM with attention for recognition, and a classifier for language identification. An EvArest dataset with 510 images for bilingual Arabic-English text from Egyptian natural scenes, annotated for bounding boxes and transcriptions. ICDAR2019 dataset includes 1200 images for multi-lingual annotations, including Arabic-English pairs. CRR: 88.9% (unified text direction improves performance). F-score: 80.5% for detection.

Arafat et al. [7] developed a deep learning-based system for Urdu text detection and recognition, extended to bilingual Urdu-English contexts. It uses Faster R-CNN for detection and a CRNN (CNN + RNN + CTC) for recognition, with implicit language identification via script-specific processing. A custom dataset consists of 2,835 images with 59,703 Urdu characters, 42,297 Urdu words, and 42,672 English characters, derived from 945 natural scene images. Supplementary datasets ICDAR2015 and CTW1500 were used for evaluation. CRR is 97.8% for Urdu and 98.7% for English. WRR: 96.5% for Urdu. The system outperforms traditional methods like MSER (70.5% CRR).

Neelotpal Chakraborty et al. [8] proposed an application of daisy descriptors for language identification in wild scene images. Firstly, RGB images are converted into grayscale, and then contrast is enhanced using an adaptive technique. After that, a Gaussian filter is applied for the removal of noise and Richardson-Lucy operation for blur removal. Further, an intensity level histogram and adaptive K-means clustering are used to separate the text and non-text candidate regions, followed by a daisy-based feature and an SVM classifier for word-level classification. The proposed algorithm was applied to different types of datasets like ICDAR2017, MLe2e, and KAIST, as well as in-house datasets, achieving precision scores of 0.788, 0.859, 0.689, and 0.837, along with recall scores of 0.7, 0.86, 0.79, and 0.89, respectively.

Ashwaq Khalil et al. [9] proposed ResNet50 and a modified EAST for Arabic, Korean, Bengali, Japanese, Chinese, and Latin scripts. The IncepText method utilizes ResNet50 for feature extraction; two layers of ResNet50 and a modified EAST model are employed for script mapping, and the max score method is applied. Furthermore, they evaluated the proposed model with datasets such as ICDAR MLT 2017, MLe2e, and Arabic-Latin, resulting in precision scores of 62.44%, 84.00%, and 83.15%, and recall scores of 54.34%, 81.00%, and 63.53%, respectively.

Zhiyun Zhang et al. [10] proposed improved script identification in natural scene text images using a CNN classifier, i.e., FAS-Res2net and Feature Pyramid Network. The proposed model employs semantic features and shallow geometric features for natural scene text images. Further, they evaluated the proposed method and achieved identification rates of 96.0% and 94.7% on public script identification datasets CVSI-2015 and SIW-13, respectively.

2. Data Collection

Bilingual Kannada and English natural scene images are not available in the literature [11]. Hence, we constructed our own real-time dataset of bilingual scene images, visiting various cities like Kalaburagi and Bidar for the collection of samples. The total dataset size is 560 scene images. Furthermore, we also collected 200 bilingual scene images from the multilingual MLe2e [13] and Char74k [14] standard datasets. While capturing the images, we used an OPPO Reno 10 mobile camera with 64 megapixels. The collected dataset has many categories such as Wall Paint, Iron, Poster, and Stone, as shown in Table 1.

Figure 1. Original Dataset of Bilingual Kannada and English Natural Scene Images

Table 1. Category of Data Samples

Category	Wall	Iron	Sign Board	Stone	Total
Low resolution	44	86	83	51	264
High resolution	129	144	187	36	496
Total	173	230	270	87	760

In Table 1 above, a distribution of images based on resolution and surface type is presented. It categorizes the images into low and high resolution across four surface types: Wall, Iron, Poster, and Stone. A total of 760 images is a mixture of all types of variations, including low quality, complex backgrounds, complicated backdrops, and perceptual distortions under various lighting conditions.

2.1 Dataset Annotation

In supervised learning, each image is to be labeled as either Kannada or English based on the text present in the image. For annotation, we employed the makesense.ai [15] tool. In the process of annotating the dataset, each script text section is marked with a rectangular bounding box with a color indication. The relevant annotation file is stored in a separate .txt file. The prefix of the image filename is the same as that of the annotation .txt file, meaning It means that both files have the same name and differ only in their extension. A sample of the Kannada and English natural scene dataset is shown in Fig. 1.

2.2 Data Validation

We validated the dataset based on the objectives of the paper. For example, each image contains text information in a bilingual format, such as Kannada and English. During the validation process, we verified that each image contains text in two different languages, and subsequently created an image annotation for training and testing the dataset to predict the text present in the scene image or underlying the given input image.

2.3 Data Augmentation

YOLOv5 employs various data augmentation techniques to improve model generalization and robustness. One of the most notable techniques is Mosaic Augmentation, which combines four images into one, helping the model detect objects at different scales. Additionally, MixUp Augmentation blends two images to enhance robustness against occlusions. HSV Augmentation modifies hue, saturation, and brightness to improve color invariance, while flipping, scaling, and rotation introduce spatial variations. Perspective transformations apply slight distortions to simulate real-time camera angles, and Cutout Augmentation removes random patches from images to help the model learn object detection despite partial occlusions.

3. Proposed Methodology

In this paper, we propose a YOLOv5 model based on CSPDarkNet53 as a fine-tuned version of YOLOv5 for bilingual Kannada and English text detection and language identification in natural scene images. Generally, the architecture of YOLO performs efficient multiple target object detection. We employed YOLOv5 for the identification of text and language present in the given input image, as shown in Fig. 1.

The YOLOv5 architecture is designed for real-time object detection. It consists of several components, each responsible for different tasks, such as feature extraction, prediction, and bounding box regression.

Backbone (Feature Extraction): YOLOv5 frequently uses convolutional neural networks (CNNs) as the basis for feature extraction from input image. The backbone network can be represented as a series of convolutional layers, followed by activation functions and pooling layers.

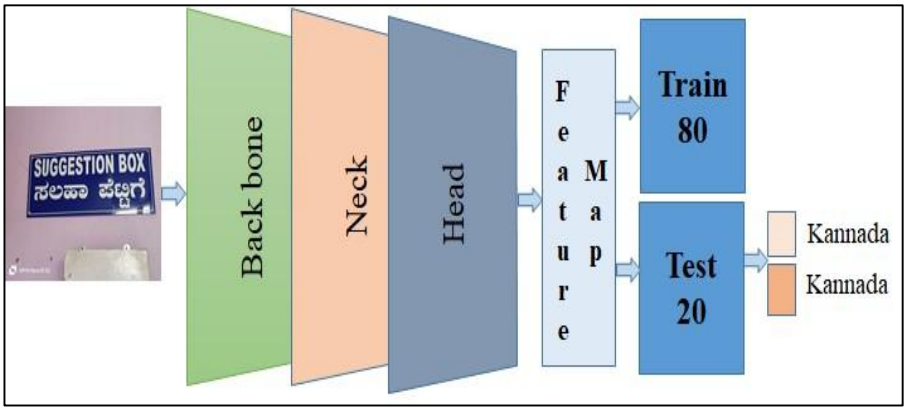


Figure 2. Proposed Methodology

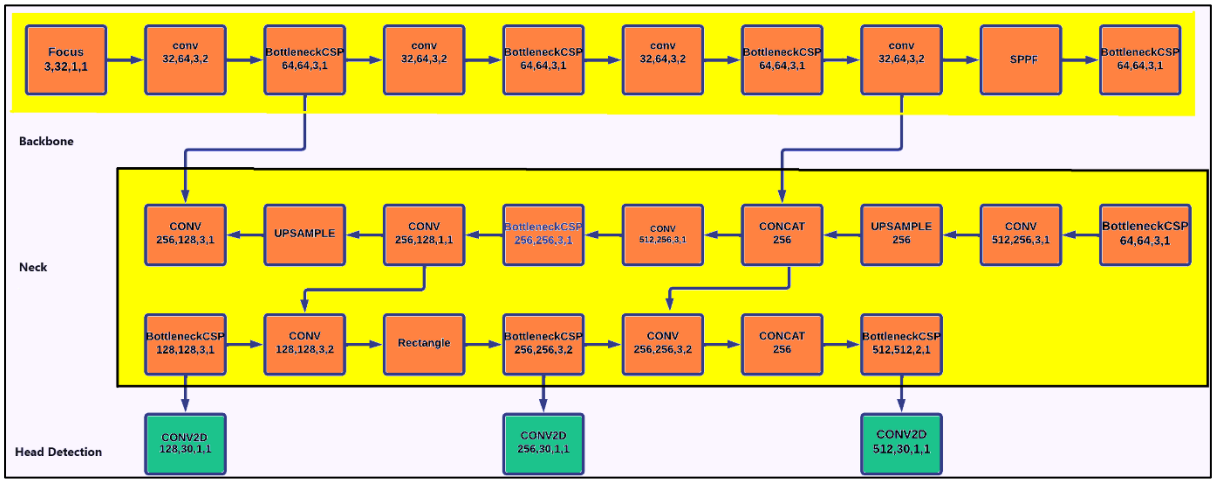


Figure 3. Fine-tuned YOLOv5 Architecture

Let X be the input image, and F_i be the function representing the $i-th$ convolutional layer equation (1):

$$F_i X = \sigma(Conv(X, W_i) + b_i \tag{1}$$

Where $Conv(X, W_i)$ is the convolution operation with weights, W_i, b_i is the bias term. σ is the activation function (e.g., ReLU). $F_i(X)$ is the output feature map after the F_{final} convolutional layer.

The final feature map F_{final} is the output of the last convolutional layer equation (2):

$$F_{final} = F_n\left(F_{n-1}\left(\ldots F_1(X)\ldots\right)\right) \tag{2}$$

Neck (Feature Pyramid Network - FPN): The neck is responsible for aggregating features at different scales to help detect objects of various sizes. This typically involves upsmpling and concatenating feature maps equation (3).

Let P_i represent the feature map at scale i:

$$P_i = Upsample\left(F_j, W_i\right) \tag{3}$$

Where $Upsample(F_i)$ is the upsampling operation, F_j is a lower-level feature map that is combined with P_i , the addition represents the feature fusion.

Head (Detection Head): The head of the YOLOv5 model predicts bounding boxes, objectness scores, and class probabilities. It does this by applying a series of convolutional layers to the aggregated feature maps equation (4).

Let H_i represent the detection head applied to the feature map P_i:

$$B_i = Conv\left(P_i, W_{bbox}\right) \tag{4}$$

Where B_i represents the bounding box predictions for scale i.

For each bounding box, YOLOv5 predicts: Coordinates, the objectness score, $P_{class}(c \mid b)$ and the class probabilities for each c

The final output for each bounding box can be represented as in equation (5):

$$Output = \left\{\hat{b}_x,\ \hat{b}_y,\ \hat{b}_w,\ \hat{b}_h,\ \hat{b}_{obj},\ \hat{b}_{class}\ (c \mid b)\right\} \tag{5}$$

Non-Maximum Suppression (NMS): After predicting multiple bounding boxes, YOLOv5 applies NMS to eliminate overlapping boxes. The NMS algorithm selects the bounding box with the highest objectness score and suppresses all others with an Intersection over Union (IoU) above a threshold.

Let $IoU\left(\mathrm{x}, \mathrm{y}\right)$ be the Intersection over Union between two boxes A and B, as shown in equation (6):

$$IoU\left(\mathrm{x}, \mathrm{y}\right) = \frac{x \cap y}{x \cup y} \tag{6}$$

If $IoU\left(\mathrm{x}, \mathrm{y}\right) > threshold$, the suppressed bounding box has the lower objectness score.

Bounding Box Prediction: In YOLO, every grid cell forecasts multiple bounding boxes.

The following defines each bounding box:

The mathematical formulation for the final bounding box coordinates is represented in equations (7), (8), (9), (10)

$$B_x = \sigma\left(t_x\right) + c_x \tag{7}$$

$$B_y = \sigma\left(t_y\right) + c_y \tag{8}$$

$$B_w = p_w e^{t_w} \tag{9}$$

$$B_h = p_h e^{t_h} \tag{10}$$

Where the coordinates of the bounding box's center in relation to the grid cell's boundaries are represented by the variables (B_x), the bounding box's anticipated width and height are denoted as B_w and B_h are offsets from the established anchor boxes. t_x, t_y, t_w *and* t_h . Here estimates for the width, height, and bounding box coordinates are provided. The grid cell's top-left coordinates are c_x and c_y. The anchor box's predetermined width and height are

denoted by p_w and p_h. Here, the sigmoid function, σ makes sure that b_x and b_y are inside the boundaries of the grid cells.

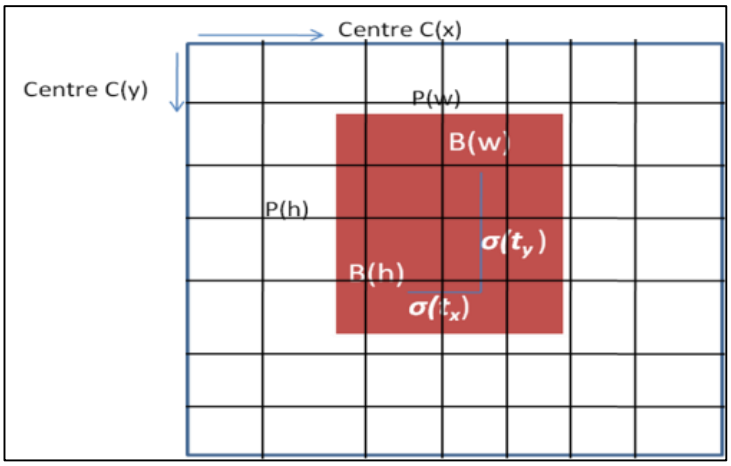


Figure 4. Boundary Box Prediction

Class Probability Prediction: For each bounding box, YOLOv5 predicts the confidence score that the bounding box contains an object, and also the class probabilities if an object is present in equation (11).

The final prediction for each class in the bounding box is:

$$P_{(class_i)} = P_{(object)} * P_{(class_i|object)} \tag{11}$$

Where, P_{object} probability that an object is present in the bounding box $c_1, c_2, \ldots, c_n$ are conditional class probabilities for n classes.

Loss Function: YOLOv5 uses a combination of three loss components are Localization loss (L1 or L2 loss) for bounding box coordinates and confidence loss for objectness score Classification loss (typically Cross-Entropy loss) for class probabilities represent in equation (12).

The total loss can be written as:

$$loss = \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(b_x - \hat{b}_y)^2 + (b_y - \hat{b}_y)^2 + (b_y - \hat{\hat{b}}_y)] + \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(b_w - \hat{b}_w)^2 + (b_h - \hat{b}_h)^2] +$$

$$\lambda_{nobj} \sum_{i=0}^{s^2} \sum_{j=0}^{B} 1_{ij}^{obj} (c_i - \hat{c}_i)^2 + \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(c_i - \hat{c}_i)^2] + \lambda_{coord} \sum_{j=0}^{B} 1_{ij}^{obj} \sum_{c\grave{o}\ classes} [P(c_i) - P(\hat{c}_i)]^2 \tag{12}$$

Where s is the grid size B is the number of bounding boxes per grid cell. 1_{ij}^{obj} is 1 if object exists in the bounding box, 0 otherwise. 1_{ij}^{nobj} is1if no object exists in the bounding box, 0 otherwise. λ_{coord} *and* λ_{nobj} are hyper parameters controlling the weight of different loss components.

Algorithm 1. Bilingual Text Detection and Language Identification

Initialize:
Img scene = *Img train*(80%)+*Img test*(20%) batch size = 16 epochs E = 100
anchor scales adjusted for small/elongated
text add *language id head* (parallel to detection head)
Training:
for i = 1 to E do
    Load bilingual train images (Kannada + English)
    Forward pass through fine-tuned YOLOv5 backbone and detection head Compute losses:
        $L_{total} = L_{box} + L_{object} + L_{Class} + L_{language}$
    Backpropagate and update parameters
    Validate using *Img test*
set end for
Save the optimal checkpoint bestweight.pt
Test:
for each *testImage* ∈ *Img test* do
    Predict $y = f(P_c, L, B_w, B_h, B_x, B_y)$
    Obtain bounding boxes with confidence scores P_c
    Assign detected word to language $L \in$ {Kannada, English}
    Display detected text region, class c, probability P_c, and language L end for

The fine-tuned YOLOv5 configuration is well-suited for text detection and language identification in natural scene image tasks, especially in bilingual contexts, due to its Focus module and flexible number of classes. The Focus module ([64, 3]) processes the input image (e.g., 640x640x3) by slicing it into a 320x320x12 feature map and applying a 3x3 convolution, preserving fine-grained details essential for detecting characters across diverse scripts (e.g., Latin, Brahmi). This lightweight approach is more efficient than the second configuration's 6x6 Conv module ([64, 6, 2, 2]), which downsamples directly and requires more computation, potentially missing subtle features in small objects like characters. The variable number of classes in the first configuration supports a wide range of character classes, accommodating multilingual datasets with diverse scripts, whereas the second configuration's fixed nc: 2 limits it to binary tasks. Pre-processing, such as normalization or augmentation, can further enhance the Focus module's ability to handle multilingual inputs, making fine-tuned YOLOv5 ideal for robust, scalable character detection across languages.

4. Experimental Results and Discussion

We employed a cross-validation technique to split the datasets into 50:50, 60:40, 70:30 and 80:20 for training and testing. The optimum parameters are as follows: Batch size = 16, Number of Epochs = 100, Adam optimizer, initial learning rate is 4, and Weights = 72 million parameters, fixed based on experimental observation.

Table 2. Performance Metrics for Different Data Splits in (percentage %)

Data Split	Precision	Recall	F1-Score	mAP50s	mAP50-95	Accuracy
50:50	74.4	68.3	71	73.7	41.6	86.6
60:40	79.4	74.0	77	79.2	45.6	90.6
70:30	83.3	73.8	78	81.0	46.1	90.7
80:20	86.8	83.4	85	85.0	50.3	94.4

The above Table 2 present evaluation metrics (Precision, Recall, F1-Score, mAP50, mAP50 95, and Accuracy) for different data splits (50:50 to 80:20). Precision vs. Recall Trade-off as the training data increases (from 50% to 80%), precision improves significantly (from 74.4 to 86.8), indicating fewer false positives. Recall also increases (from 68.3 to 83.4), showing that more actual detected text and language identification. The 80:20 split achieves the best balance between both, with high recall and precision, leading to the highest F1-score (85). F1-Score as a Balanced Measure. The F1-score, which balances precision and recall, improves from 71 (50:50) to 85 (80:20), indicating better overall detection performance as more training data is used. mAP Performance (Localization and Detection Quality) mAP50 (mean Average Precision at IoU 0.5) improves steadily from 73.7 to 85.0, showing that as more training data is available, the model detects the text and language identification more accurately. mAP50-95 (a stricter measure considering multiple IoU thresholds) increases from 41.6 to 50.3, suggesting that the model's localization improves with training data. We observed in the above table that the optimum accuracy is 94.4, with 80% of training and 20% of testing data sets. The higher the value of precision, the lower the recall value. It indicates that performance is better with corresponding data set and its cross validations. To test the efficacy of the model, the trained model is tested on several real-time Natural scene images.

The difference between mAP@50 (lenient threshold) and mAP@50:95 (more stringent criterion) underscores the model's deficient localization capacity. The detector can generally identify the presence of Kannada and English text, but it has difficulty accurately aligning bounding boxes.

Statistical validation: The performance evaluation metrics are validated using the confusion matrix as shown in Table 3

Precision: Measures the proportion of detected objects that are actually correct (i.e., true positives over all positive detections). In object detection, a high precision is crucial when the cost of false positives is high.

$$Precesion = \frac{Tp}{Tp + Fn} \tag{13}$$

Recall: Demonstrates the percentage of real items found, or true positives over the total of false negatives and true positives.

$$\mathrm{Re}\,call = \frac{Tp}{Tp + Fn} \tag{14}$$

F1 Score: This harmonic mean of precision and recall provides a single metric to evaluate the balance between the two. Its absence could be explained by a desire to present the

individual components rather than a composite score, thereby offering more detailed insights into where the model performs well or needs improvement.

$$F1 = 2\frac{\text{Precesion*Recall}}{\text{Pr}\,ecesion + \text{Re}\,call} \tag{15}$$

Mean Average Precision: A measure called mAP@0.5, or mAP@0.5-9.5 at an Intersection over Union (IoU) threshold of 0.5, is used to assess how well object identification models perform. It shows how well the model detects items that have at least 50% overlap with the ground truth.

$$mAP = \frac{1}{n}\sum_{i=0}^{n} n \tag{16}$$

Accuracy: While less commonly used in pure object detection due to the imbalance between object and background classes, it can still provide a high-level overview of overall performance, especially when combined with other metrics.

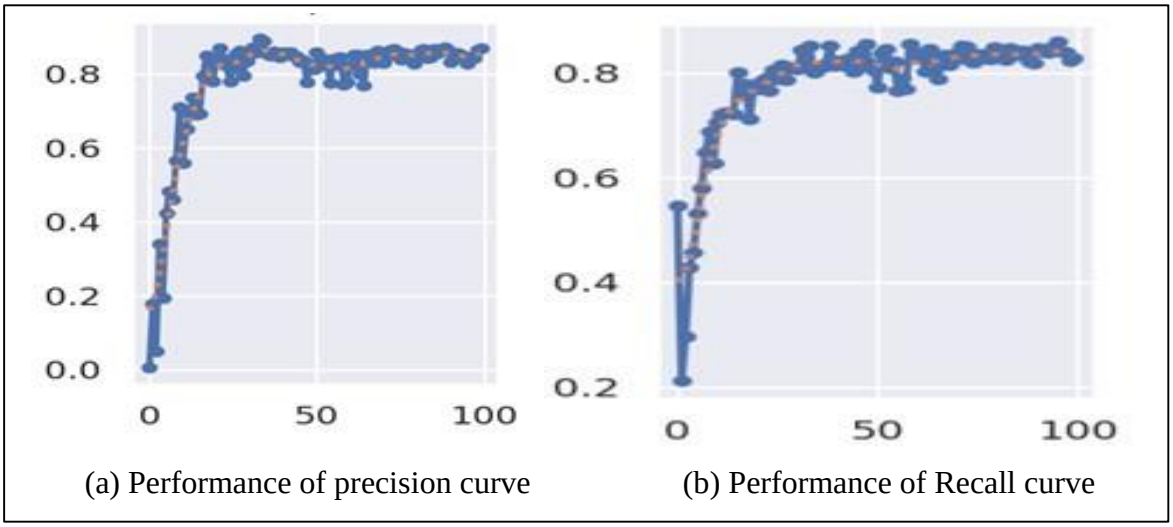
$$Accuracy = \frac{Tp + Tn}{TotalNumber\,of\ \text{Prediction}} \tag{17}$$

Table 3. Confusion Matrix for 80:20

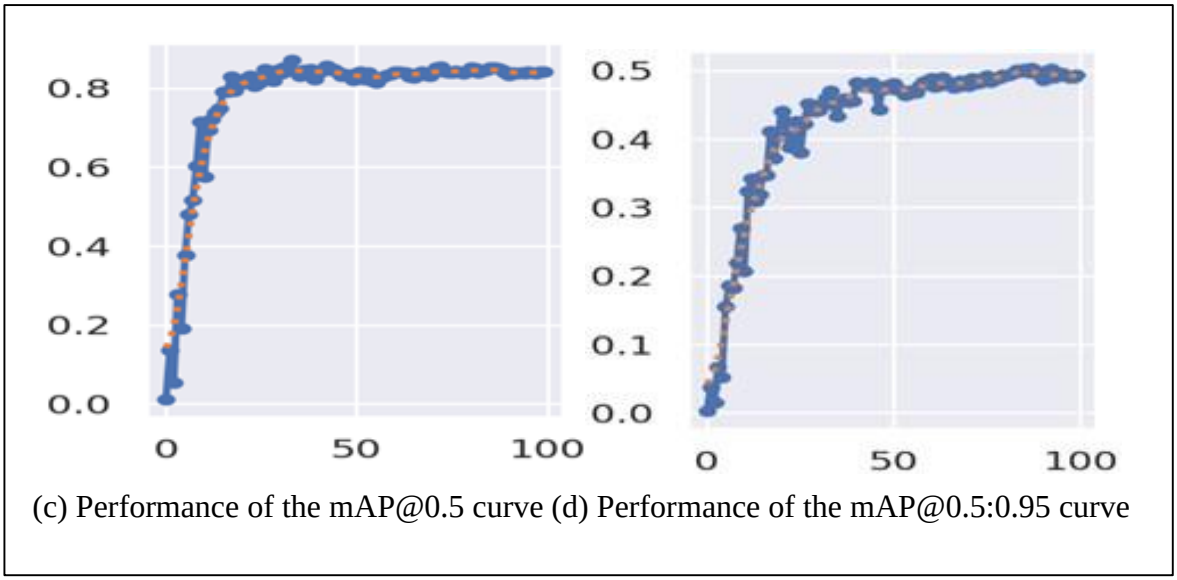
	Kannada	**English**	**Background**
Kannada	0.85	0.04	0.45
English	0.05	0.90	0.55
Background	0.10	0.06	0.00

Table 4. Comparative Analysis of Existing Method Vs Proposed Method

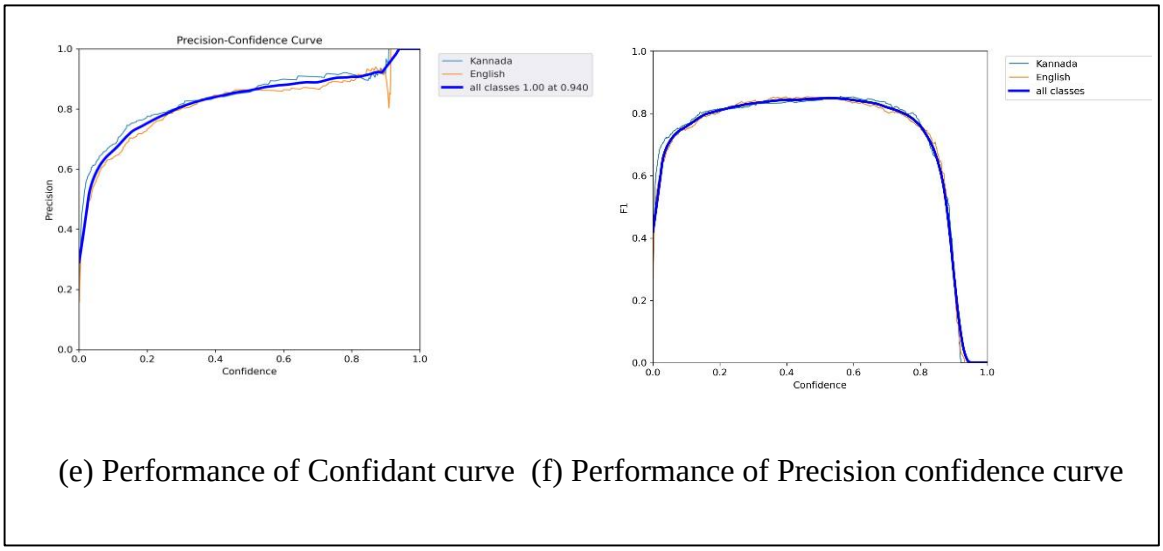
Paper	**Techniques**	**Dataset**	**Size**	**Precision**	**Recall**
Alex Noel et al.[4]	Faster R -CNN	MSRA-TD500	500	85.08%	59.60%
Proposed Model	YOLOv5	Own Dataset	760	86.8%	83.4%



(a) Performance of precision curve (b) Performance of Recall curve



(c) Performance of the mAP@0.5 curve (d) Performance of the mAP@0.5:0.95 curve



(e) Performance of Confidant curve (f) Performance of Precision confidence curve

Figure 5: Performance Metrics for the Bilingual Fine-tuned YOLOv5model: (a) Performance of Precision, (b) Performance of Recall, (c) Performance of mAP@0.5, (d) Performance of mAP@0.5:0.95, (e) Performance of Confidant curve, (f).Performance of Precision Confidence Curve F1

The comparative analysis between the proposed YOLOv5-based model and the existing Faster R-CNN-based model by Alex Noel et al. [4] highlights significant improvements in text detection performance. The existing model, which was evaluated on the MSRA-TD500 dataset containing 500 images, achieved a precision of 85.08% and a recall of 59.60%. In contrast, the proposed model, trained on a newly collected bilingual Kannada-English dataset with 760 real-time images, demonstrated superior performance with a precision of 86.8% and a significantly higher recall of 83.4%. The higher recall indicates that the proposed model misses fewer text instances, making it more effective for real-world applications. Additionally, YOLOv5, being a one-stage detection model, offers better efficiency and faster processing compared to the two-stage Faster R-CNN approach. This makes the proposed model more suitable for real-time applications such as automated signboard reading, language translation, and assistive technologies. The use of a larger and more diverse dataset further contributes to the improved

accuracy of the proposed model. Overall, these results demonstrate that the YOLOv5-based approach is a more effective solution for bilingual scene text detection and language identification.

Figure 6. Recall vs Training Progress

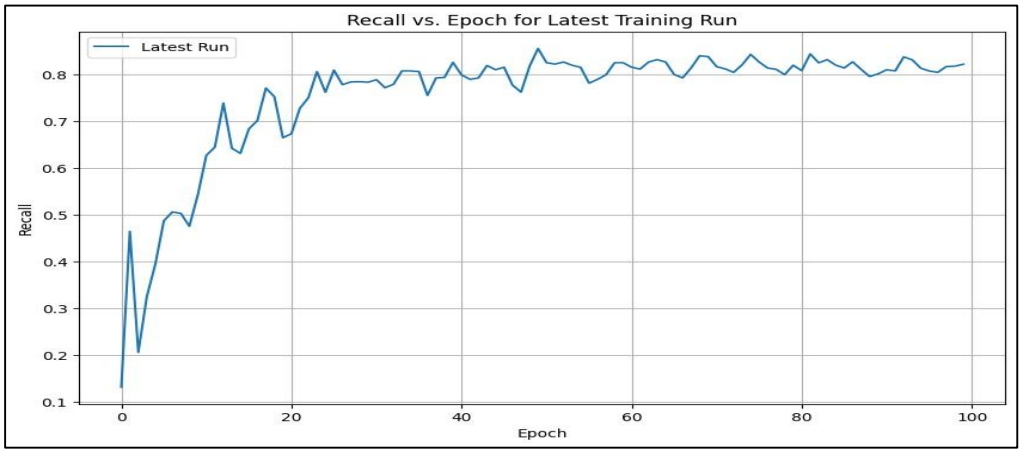


Figure 7. mAP vs Training Progress

Figure 8. Sample Image of Text Detection and Language Identified

4.1 Limitation of YOLOv5

YOLOv5, while powerful, has limitations including struggles with detecting small objects, dense scenes, and maintaining high accuracy in complex environments. It also requires a large amount of training data and can be computationally intensive.

5. Conclusion

In this paper, a fine-tuned deep learning model for natural scene text detection and language identification is proposed. Here, we used a fine-tuned deep learning architecture, YOLOv5, and a novel real-time dataset comprising 760 bilingual scene text images collected from the Kalaburagi and Bidar cities in the state of Karnataka. The proposed deep learning approach effectively integrates text detection with language identification. This is supported by experimental results, demonstrating a precision rate of 86.8%, a recall rate of 83.4%, an F1 score of 85%, and an accuracy of 94.4%. The proposed model indicates robust performance across diverse scenarios with complex backgrounds; however, the small size of text present in the images is a limitation of the YOLOv5 model. The proposed model is also compared with existing techniques found in the literature to assess its robustness. The proposed model outperforms existing traditional techniques for bilingual natural scene text detection and language identification. In the future, we will extend this work to South Indian natural scene images in terms of bilingual and multilingual capabilities to meet the requirements of a multilingual OCR system for natural scene images.

Acknowledgment

The authors are grateful to KSTePS DST, Government of Karnataka, India, for their support to carry out the Research work.

References

[1] Khan, Nisar, Riaz Ahmad, Khalil Ullah, Siraj Muhammad, Ibrar Hussain, Ahmad Khan, Yazeed Yasin Ghadi, and Heba G. Mohamed. "Robust arabic and pashto text detection in camera-captured documents using deep learning techniques." IEEE Access 11 (2023): 135788-135796.

[2] Gomez, Lluis, and Dimosthenis Karatzas. "A fine-grained approach to scene text script identification." In 2016 12th IAPR workshop on document analysis systems (DAS), IEEE, 2016, 192-197.

[3] Maheshwari, Karan, Alex Noel Joseph Raj, Vijayalakshmi GV Mahesh, Zhemin Zhuang, Elizabeth Rufus, Palaiahnakote Shivakumara, and Ganesh R. Naik. "Bilingual text detection in natural scene images using invariant moments." Journal of Intelligent & Fuzzy Systems 37, no. 5 (2019): 6773-6784.

[4] Joseph Raj, Alex Noel, Chen Junmin, Ruban Nersisson, Vijayalakshmi GV Mahesh, and Zhemin Zhuang. "Bilingual text detection from natural scene images using faster R-CNN and extended histogram of oriented gradients." Pattern Analysis and Applications 25, no. 4 (2022): 1001-1013.

[5] Chandio, Asghar Ali, Md Asikuzzaman, Mark Pickering, and Mehwish Leghari. "Cursive-text: a comprehensive dataset for end-to-end Urdu text recognition in natural scene images." Data in brief 31 (2020): 105749.

[6] Albalawi, Bayan M., Amani T. Jamal, Lama A. Al Khuzayem, and Olaa A. Alsaedi. "An End-to-End Scene Text Recognition for Bilingual Text." Big Data and Cognitive Computing 8, no. 9 (2024): 117.

[7] Arafat, Syed Yasser, and Muhammad Javed Iqbal. "Urdu-text detection and recognition in natural scene images using deep learning." IEEE Access 8 (2020): 96787-96803.

[8] Chakraborty, Neelotpal, Agneet Chatterjee, Pawan Kumar Singh, Ayatullah Faruk Mollah, and Ram Sarkar. "Application of daisy descriptor for language identification in the wild." Multimedia Tools and Applications 80, no. 1 (2021): 323-344.

[9] Khalil, Ashwaq, Moath Jarrah, Mahmoud Al-Ayyoub, and Yaser Jararweh. "Text detection and script identification in natural scene images using deep learning." Computers & Electrical Engineering 91 (2021): 107043.

[10] Zhang, Zhiyun, Hornisa Mamat, Xuebin Xu, Alimjan Aysa, and Kurban Ubul. "FAS-Res2net: An improved res2net-based script identification method for natural scenes." Applied Sciences 13, no. 7 (2023): 4434.

[11] Hangarage, Venkata, and Gururaj Mukarambi. "Text Localization and Enhancement of Mobile Camera based Complex Natural Bilingual Text Scene Images." Procedia Computer Science 235 (2024): 2353-2361.

[12] Dhandra, B. V., Satishkumar Mallappa, and Gururaj Mukarambi. "Script identification of camera based bilingual document images using SFTA features." International Journal of Technology and Human Interaction (IJTHI) 15, no. 4 (2019): 1-12.

[13] Gomez, Lluis. (2016). MLe2e multi-lingual end-to-end dataset.

[14] de Campos, T. E., Babu, B. R., Varma, M. Character recognition in natural images. International conference on computer vision theory and applications(2009),1, SCITEPRESS, 273-280.

[15] Makesense AI. (n.d.). Retrieved from https://www.makesense.ai/