IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 6, June 2025

DOI: 10.17148/IJARCCE.2025.14694

Lightweight Script Classification for Multilingual Scene Text Recognition Using MobileNetV2

Vishnuvardhan Atmakuri[1], M. Dhanalakshmi[2]

Research Scholar in JNTUH- Jawaharlal Nehru Technological University Hyderabad, India[1]

Professor, Dept of Information Technology, Deputy Director, DILT, JNTUH, Hyderabad, India[2]

Abstract: In multilingual scene text recognition, accurate identification of the script used in each text region is essential before applying language-specific OCR. This paper proposes a lightweight script classification module based on MobileNetV2 [1], integrated into a broader Telugu scene text recognition pipeline. The system first detects word-level text regions using an enhanced EAST detector and then classifies each region into one of three script classes Telugu, English, or Hindi. The proposed classifier leverages transfer learning, efficient preprocessing, and a balanced dataset augmented to address class imbalance. Experimental results show that the classifier achieves a high overall accuracy of 94.81%, with minimal inter-script confusion, even in visually cluttered scenes. Qualitative examples and a detailed confusion matrix validate the model's robustness and generalizability. This approach demonstrates how lightweight deep learning models can be effectively used in real-world OCR systems, particularly for Indian languages. Future directions include expanding script coverage, enabling handwritten text recognition, and integrating the module into an end-to-end OCR pipeline.

Keywords: Script Classification, MobileNetV2, Multilingual Scene Text, Transfer Learning, OCR Pipeline.

I. INTRODUCTION

Accurate recognition of multilingual text in natural scene images critically depends on the system's ability to identify the script of each detected text region before applying language-specific OCR models [2], [8], [9]. This task becomes particularly challenging in real-world scenarios where multiple scripts may co-occur, and visual noise or background clutter complicates recognition [3]. To address this issue, the proposed system introduces a script classification module that operates immediately after the text detection stage. Each word-level text region detected by the system is passed through a lightweight MobileNetV2-based convolutional neural network, which classifies the script into one of three categories: Telugu, Hindi, or English. This intermediate classification enables the selective use of script-specific OCR engines tailored for the identified script, thereby improving overall recognition accuracy and system robustness. By positioning this module between detection and recognition, the system enhances its adaptability to complex multilingual environments. The high-level workflow of the proposed pipeline, showing the integration of text detection and script classification, is depicted in Figure 1. This paper details the methodology, implementation, and performance evaluation of the classification module using both qualitative and quantitative analysis.

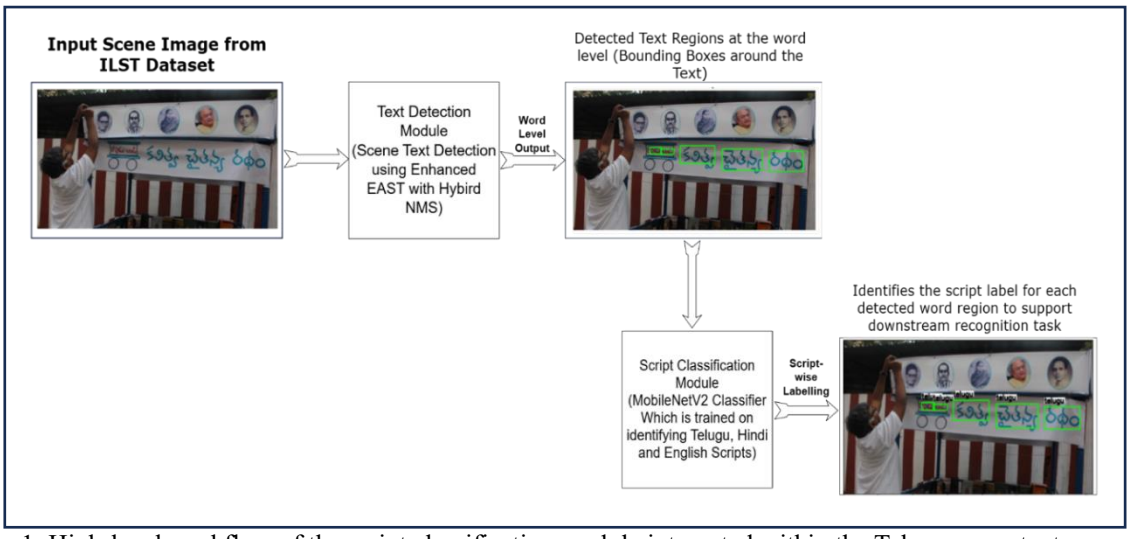

Figure 1: High-level workflow of the script classification module integrated within the Telugu scene text recognition pipeline

DOI: 10.17148/IJARCCE.2025.14694

To address the challenge of efficient and accurate script identification in multilingual scene text, this paper proposes a lightweight classification framework that integrates seamlessly into the overall text recognition pipeline. The key contributions of this work are as follows: (i) a high-level system architecture combining enhanced EAST [4] for text detection with a MobileNetV2-based classifier for script identification; (ii) the development of a balanced, word-level dataset comprising Telugu, English, and Hindi scripts, including extensive augmentation for underrepresented classes; (iii) a detailed implementation of the script classification module using transfer learning and efficient preprocessing strategies; and (iv) comprehensive experimental validation demonstrating the classifier's robustness across visually similar scripts in real-world multilingual scene images. Together, these contributions establish a scalable approach to improving script-aware OCR in complex visual environments.

The remainder of this paper is organized as follows. Section 2 presents a review of related work in the area of script identification in natural scene images. Section 3 describes the proposed methodology, including the overall pipeline, detection module, and the MobileNetV2-based script classification algorithm. Section 4 details the dataset preparation and augmentation strategies used to address class imbalance. Section 5 discusses the experimental results, including confusion matrix analysis and performance metrics. Finally, Section 6 concludes the paper with key findings and potential directions for future work.

II. RELATED WORK

In [3] Naosekpam and Sahu proposed WAFFNet, a deep learning-based framework specifically designed for script identification in natural scene images featuring regional Indian scripts. Their approach integrates convolutional features extracted from the first four blocks of a pre-trained VGG16 model with handcrafted Local Binary Pattern (LBP) features. These features are combined using a spatial attention mechanism that emphasizes semantically significant regions in the image. WAFFNet achieves a balance between model efficiency (~8 million parameters) and classification performance. To address the challenge of underrepresented scripts, the authors developed the IIITG-STLI2023 dataset, which includes word-level scene text images across seven Indian scripts: English, Hindi, Bengali, Manipuri, Malayalam, Telugu, and Kannada. Experimental results demonstrated that WAFFNet surpassed existing methods, achieving average accuracies of 92.13% on IIITG-STLI2023, 98.0% on MLe2e, and 92.43% on SIW-13. The model exhibited strong generalization capability, particularly for structurally similar scripts such as Telugu and Kannada, highlighting its effectiveness for multilingual OCR in complex street scene environments.

In [2], Bhunia et al. proposed an advanced deep learning architecture for script identification in natural scene images and video frames, incorporating a combination of Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) units, and a feed-forward attention mechanism. The framework processes image patches extracted from textual regions, where CNNs capture localized visual features and LSTMs model sequential spatial relationships. An attention mechanism is applied to assign higher importance to informative patches, generating local features, while a global feature is derived from the LSTM's final cell state. These features are dynamically fused through a weighted scheme to perform script classification. The model was evaluated on multiple public datasets, including SIW-13, CVSI-2015, ICDAR-2017, and MLe2e, and consistently outperformed traditional baselines such as LBP, CRNN, ECN, and MSPN. The architecture achieved notable accuracy rates of 96.5% on SIW-13 and 97.75% on CVSI-2015, demonstrating strong recognition performance, particularly for South Indian scripts like Telugu, Kannada, and Tamil. This work was the first to apply attention mechanisms in the context of scene text script identification, significantly enhancing robustness to background clutter and inter-script similarities.

In [5] presented a deep learning framework tailored for identifying scripts in handwritten word images across 11 different Indic languages. The approach avoids reliance on handcrafted features by employing ten separate Convolutional Neural Networks (CNNs), each trained on inputs derived from both spatial domain images and Haar wavelet-transformed frequency representations at varying scales (32×32, 48×48, and 128×128). Each CNN outputs a 1024-dimensional feature vector, and the concatenation of all outputs yields a 10240-dimensional representation. This high-dimensional feature vector is subsequently passed to a Multi-Layer Perceptron (MLP) for final classification. The model was evaluated on the PHD Indic 11 dataset, which contains 11,000 word-level images, and it achieved a peak accuracy of 94.73%, outperforming established baselines such as AlexNet (92.14%) and LeNet (82%). This work provides a strong benchmark for handwritten script identification in multilingual contexts, particularly for structurally diverse scripts including Telugu, Kannada, Urdu, and Roman.

In [12], Dhanya et al. proposed a script identification technique tailored for printed bilingual documents, specifically focusing on Roman and Tamil scripts. Their approach includes two complementary methods for word-level script classification. The first method segments each word into three spatial zones—upper, middle, and lower—and uses the

DOI: 10.17148/IJARCCE.2025.14694

distribution of ON-pixels and character density across these zones as distinguishing features. The second method applies directional energy analysis using Gabor filters to capture orientation-based differences between scripts. A combination of these features was classified using SVM and k-NN classifiers, achieving recognition accuracy above 96% with Gabor-based features. This work demonstrates early yet effective strategies for printed script identification, providing a solid foundation for zone-wise and orientation-sensitive classification in multilingual documents.

In [13], Pal and Chaudhuri presented a decision-tree-based approach for identifying scripts in Indian documents, focusing on Devanagari, Bengali, and Roman scripts. Their method exploits prominent script-specific features such as the presence of the horizontal line (shirorekha), ascender-descender patterns, and character stroke distributions. The work emphasizes line-level script recognition using structural heuristics, allowing effective classification even in noisy or degraded documents. While their approach is geared toward printed documents with mono-script lines, the incorporation of stroke-level details offers valuable insights for enhancing script identification in multilingual OCR systems.

III. METHODOLOGY

The overall architecture of the proposed multilingual scene text recognition system is depicted in Figure 1, which illustrates the sequential integration of text detection and script classification modules. Given an input natural scene image, the system first detects word-level text regions using an enhanced version of the Efficient and Accurate Scene Text (EAST) detector. These detected word crops are then passed to a lightweight MobileNetV2-based classifier, which predicts the script label Telugu, Hindi, or English for each word region. This intermediate classification step enables the downstream use of script-specific OCR engines, thereby improving the accuracy and adaptability of the recognition process in multilingual settings.

3.1 Text Detection Module

The text detection stage employs an enhanced EAST detector modified with a hybrid Non-Maximum Suppression (NMS) strategy that combines Soft NMS [6] and Adaptive NMS [7] techniques. This augmentation improves bounding box localization in cases of overlapping text and complex backgrounds. The detector outputs rotated quadrilateral bounding boxes for word-level text regions, which are then extracted for subsequent classification. A brief overview of the detection algorithm is presented in Algorithm 1, outlining the use of score maps, geometry predictions, and refined box suppression.

Algorithm 1: Telugu Scene Text Detection Using Enhanced EAST with Soft + Adaptive NMS

Input:

I → RGB input image (width and height must be multiples of 32 pixels)

X → XML file containing ground truth bounding box coordinates and word text in an image

$\mathbf{B_{GT}}$**=**$\{(\mathbf{x_{min}}, \mathbf{y_{min}}, \mathbf{x_{max}}, \mathbf{y_{max}})\}$ → Ground truth bounding boxes extracted from X

$\boldsymbol{\sigma}$ → Soft NMS decay factor (0.5 in the proposed approach)

T → Confidence threshold (0.01 in the proposed approach)

$\boldsymbol{\alpha, \beta}$ → Adaptive suppression parameters

Output:

$\mathbf{B_{final}}$**:** Suppressed bounding box coordinates

Procedure:

1. Preprocessing:

a. Read input image I and corresponding XML file X

b. Extract ground truth bounding boxes B_{GT} from X where

$$B_{GT} = \{(x_{min}, y_{min}, x_{max}, y_{max})\}$$

2. Feature Extraction:

a. Pass I through the Multi-channel Fully Convolutional Network (FCN)

Generate Output:

$$G = \{(x_{top}, y_{top}, x_{bottom}, y_{bottom})\}$$

Where G represents the predicted geometry map

b. Compute the confidence score map S with values in the range [0,1]

3. Bounding Box Generation:

a. Iterate over each predicted region in G

b. Assign confidence scores S_i to each bounding box

4. Apply Hybrid Non-Maximum Suppression (Soft + Adaptive NMS):

a. Given bounding boxes $B = \{B_1, B_2, \ldots, B_n\}$ and corresponding scores S

Compute the IoU (IoU_{ij}) between bounding boxes B_i and B_j

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 6, June 2025

DOI: 10.17148/IJARCCE.2025.14694

$$IoU_{ij} = \frac{|B_i \cap B_j|}{|B_i \cup B_j|}$$

b. Compute the Adaptive Suppression Threshold:

$$T_{adaptive} = \alpha + \beta \cdot \frac{\min(w_i, h_i)}{\max(w_i, h_i)}$$

c. Compute the Soft NMS decay function:

$$S_j = S_j \cdot e^{-(IoU_{ij}^2)/\sigma}$$

d. Retain bounding boxes where $S_j < T_{adaptive}$

5. Visualization & Output:

Overlay refined bounding boxes B_{final} onto I

Display the result image with detected bounding boxes

3.2 Script Classification Module

Each detected word-level image is passed through the script classification module, whose pipeline is illustrated in Figure 2. The classification framework is built on MobileNetV2, a lightweight convolutional neural network optimized for resource-efficient inference [12]. The module consists of several stages, including preprocessing, feature extraction, classification via a fully connected layer, softmax probability computation, and final label prediction. This design ensures fast and accurate script identification while avoiding the computational complexity of recurrent or transformer-based models.

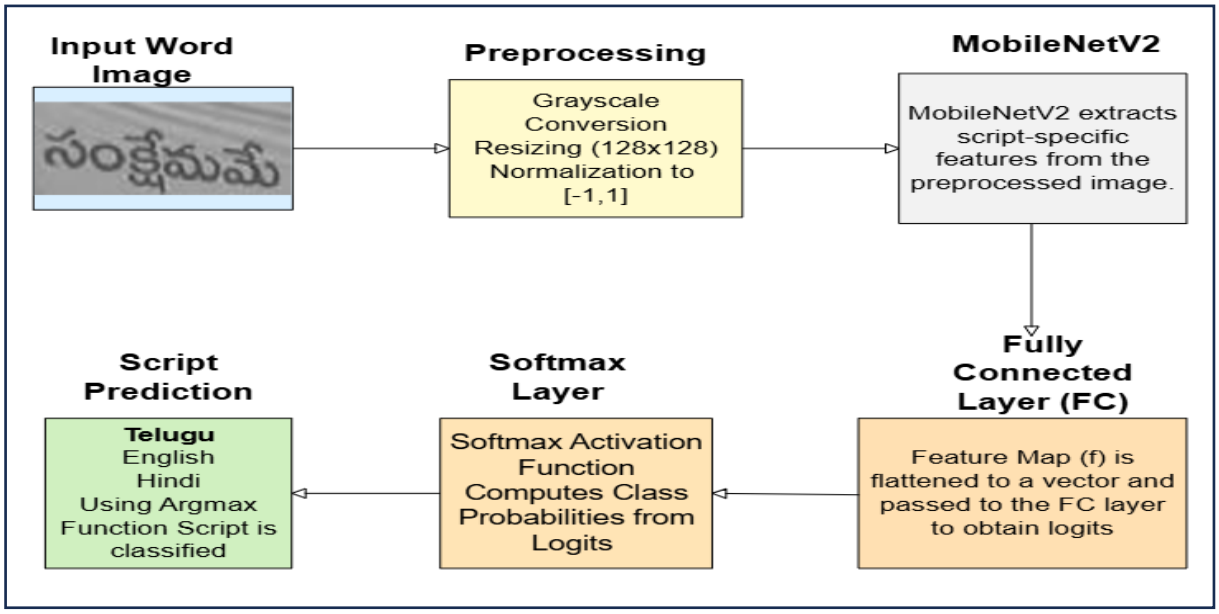

Figure 2: Script Classification Pipeline Using MobileNetV2

The classification process is formally described in Algorithm 2. The input image is first resized and normalized. Deep features are extracted using MobileNetV2's depthwise and pointwise convolutional layers. These features are passed through a fully connected layer to produce logits, which are then converted into class probabilities using the softmax function. The class with the highest probability is selected as the predicted script label. This strategy leverages the transferability of ImageNet-pretrained models for improved generalization, as demonstrated in [13].

Algorithm 2: Script Classification Algorithm based on MobileNetV2

Input:

I_{crop} ← Cropped word-level image region

M ← Trained MobileNetV2 script classification model

C ← Set of script classes {Telugu, English, Hindi}

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 6, June 2025

DOI: 10.17148/IJARCCE.2025.14694

Output:
L_{pred} ← Predicted script label

Procedure:

1. **Preprocess I_{crop}:**

a. Resize I_{crop} to fixed dimensions (128x128)

b. Convert to grayscale (1 channel)

c. Normalize pixel values to [-1,1]

2. Feature Extraction:

Extract deep features f using the convolutional layers of MobileNetV2 which includes:

- Depth wise Convolutions: Captures the features like edges, curves and character strokes by applying filters channel wise
- Point wise Convolutions: Combines the extracted features across channels to capture script-specific features at the word level

3. Classification Layer:

Pass the feature vector f into a final fully connected (FC) layer where the linear transformation is applied to obtain the logits:

$$\boldsymbol{z = W \cdot f + b}$$

Where

- f: the input feature vector (flattened)
- W, b : Weights and bias of the final FC layer
- z : Logits represent the raw prediction scores for each script class before applying the SoftMax function.

4. Softmax Probability Computation:

Compute class probabilities using the softmax activation function

$$\boldsymbol{P = softmax(z)}$$

Where

- z: Logits obtained from the Fully Connected layer
- P: Probability distribution over script classes (eg: Telugu, English and Hindi)

5. Prediction:

Assign the predicted label by selecting the class with highest probability

$$\boldsymbol{L_{pred} = argmax(P)}$$

Where

- P : Vector of Probabilities for each class
- $argmax(P)$: Finds the index of the class with highest probability
- That index corresponds to the predicted script label.

6. Return the Predicted Script Label:

Output the final predicted class label L_{pred} corresponding to the highest Probability

3.3 Implementation Details

The script classification module was implemented using the PyTorch deep learning framework in a Google Colab environment with a CPU-based runtime. Image preprocessing was performed using the Torchvision and OpenCV libraries. Each word-level image was resized to 128×128 pixels, converted to grayscale, and normalized to the range [−1,1] before being converted into tensors suitable for input into the model. MobileNetV2 was initialized with pretrained weights from ImageNet. The final classification layer was replaced with a fully connected layer containing three output neurons corresponding to the Telugu, English, and Hindi script classes. The model was trained using a categorical cross-entropy loss function and optimized using the Adam optimizer with a learning rate of 0.0005. Training was conducted for 10 epochs with a batch size of 32.

The dataset used for training comprised 1367 Telugu, 657 English, and 308 Hindi word-level images. Representative examples from the dataset, showcasing word-level image crops across the three script classes, are illustrated in Figure 3. The Hindi class was initially underrepresented but was augmented [14] using geometric and photometric transformations to improve class balance. The model and class label mappings were saved upon training completion to facilitate integration into the larger OCR pipeline.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 6, June 2025

DOI: 10.17148/IJARCCE.2025.14694

Figure 3: Sample word images from scene text, showing Telugu (top), English (middle), and Hindi (bottom) scripts used for script classification.

IV. EXPERIMENTAL RESULTS

The effectiveness of the proposed script classification module was evaluated using a dataset comprising word-level images of Telugu, English, and Hindi scripts, extracted from real-world scene images. The evaluation focused on both qualitative visualizations and quantitative performance metrics to assess the robustness and generalization capability of the model under varied lighting conditions, font styles, and background complexities.

4.1 Qualitative Results

To visually validate the performance of the script classifier, several multilingual scene images were processed through the detection and classification pipeline. Figure 4 presents example outputs showing bounding boxes and the corresponding script labels. The classifier demonstrates strong alignment with ground truth across diverse visual conditions, successfully distinguishing between scripts even when multiple scripts co-occur in the same image.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 6, June 2025

DOI: 10.17148/IJARCCE.2025.14694

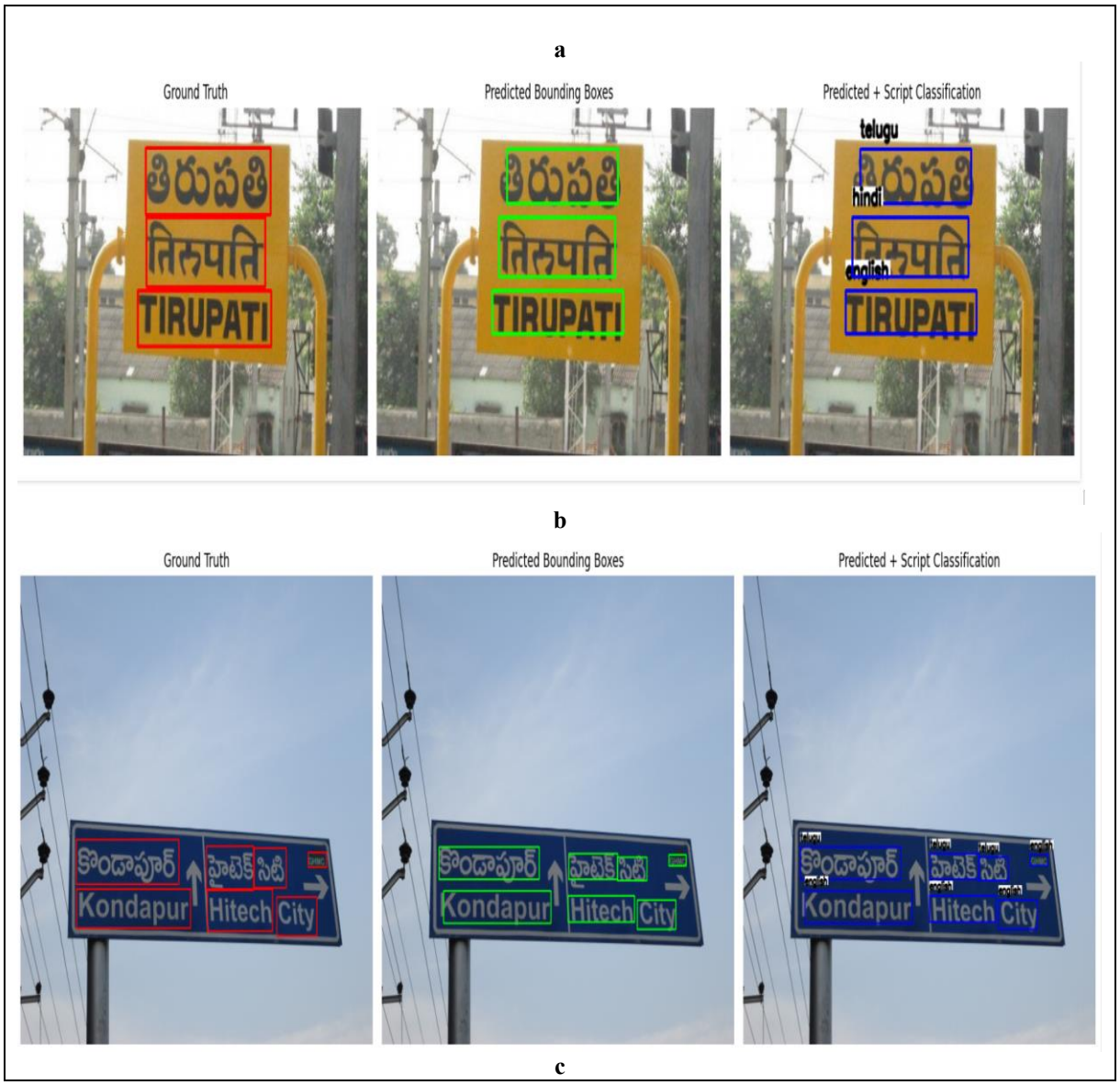

Figure 4(a, b, c): Visual comparison of ground truth, predicted text regions, and script classification in multilingual scene images

4.2 Confusion Matrix Analysis

The confusion matrix shown in Figure 5 presents the classification results of the script identification model across three script classes: Telugu, English, and Hindi. Out of 1367 Telugu instances, the classifier correctly identified 1297, while 45 were misclassified as English and 25 as Hindi. For the 657 English samples, 620 were correctly classified, with 19 misclassified as Telugu and 18 as Hindi. The Hindi class, which contained 308 instances, achieved 294 correct predictions, while 8 were incorrectly labeled as Telugu and 6 as English. The matrix highlights the classifier's strong performance overall, with minimal confusion between structurally similar scripts. These results reflect the effectiveness of the MobileNetV2-based script classification module in handling complex multilingual scene images.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 6, June 2025

DOI: 10.17148/IJARCCE.2025.14694

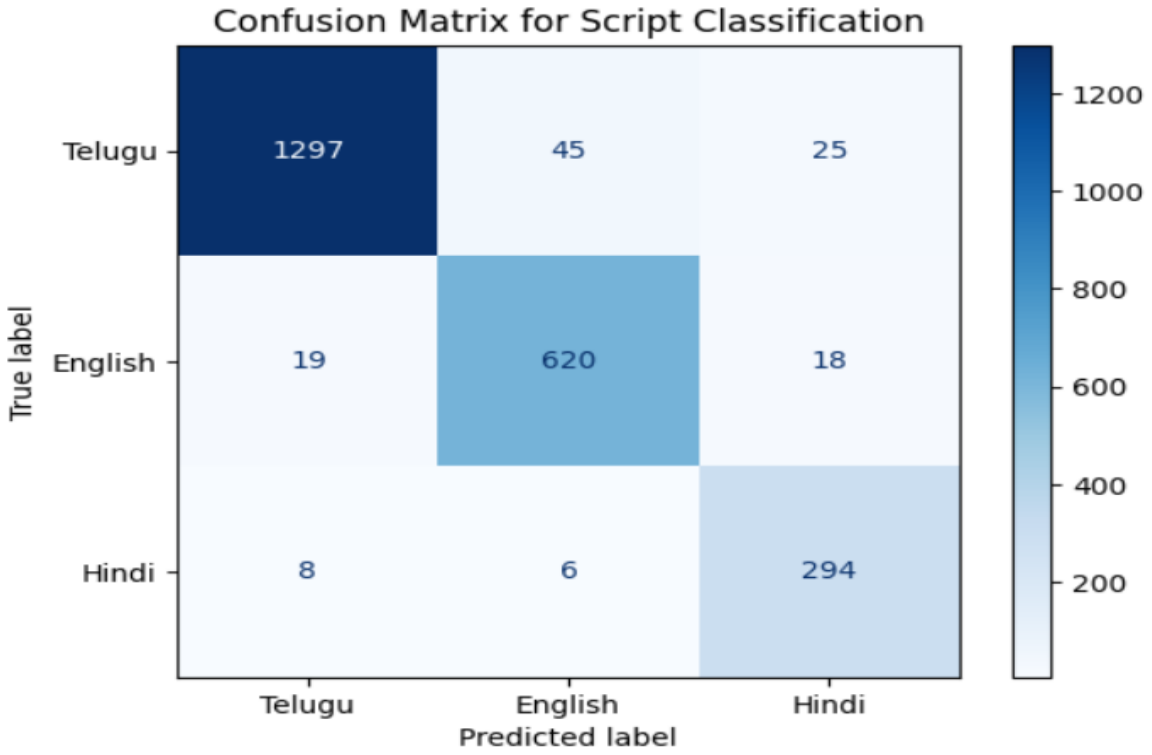

Figure 5: Confusion Matrix illustrating the performance of the proposed script classification Module using MobileNetV2

4.3 Performance Metrics

To quantitatively assess the performance of the proposed script classification module, standard evaluation metrics were computed, including Precision, Recall, F1-Score, and Overall Accuracy. These metrics offer a detailed view of the model's ability to correctly classify word-level text regions from natural scene images. Using the confusion matrix shown in Figure 5, the performance metrics were computed for each script class. As shown in Table 1 The script classification model achieved an overall accuracy of 94.81%, demonstrating its robustness [15] in identifying scripts from natural scene images. Class-wise performance analysis shows that the model attained a precision-recall of 0.97 / 0.94 for Telugu, 0.92 / 0.94 for English, and 0.87 / 0.95 for Hindi. These results confirm that the classifier performs consistently across scripts, including the previously underrepresented Hindi class, and validates the effectiveness of the proposed approach in real-world multilingual scenarios.

Table 1: Performance Metrics for Script Classification

Metric	Formula	Interpretation	Metric Computation
Precision	$\frac{TP}{TP+FP}$	Out of all predicted positives, how many were correct	$Telugu: \frac{1301}{1301+32} \approx 0.9796$ $English: \frac{624}{624+46} \approx 0.9239$ $Hindi: \frac{290}{290+39} \approx 0.8724$

DOI: 10.17148/IJARCCE.2025.14694

Recall	$\frac{TP}{TP + FN}$	Out of all actual positives, how many were detected	$Telugu: \frac{1301}{1301+66} \approx 0.9487$ $English: \frac{624}{624 + 46} \approx 0.9436$ $Hindi: \frac{290}{290 + 18} \approx 0.9545$
F1 − Score	$\frac{2 * Precision * Recall}{Precision + Recall}$	Harmonic mean of Precision and Recall	$Telugu: \approx 0.9639$ $English: \approx 0.9336$ $Hindi: \approx 0.9116$
Accuracy	$\frac{Total\ Correct\ Predictions}{Total\ Predictions}$	Overall correctness of predictions	$\frac{1297 + 620 + 294}{2332} = \frac{2211}{2332} \approx 0.9481$

V. CONCLUSION & FUTURE SCOPE

This paper presented a lightweight and efficient script classification module integrated into a multilingual scene text recognition pipeline. By employing a MobileNetV2-based architecture, the proposed system effectively identifies the script Telugu, English, or Hindi of word-level text regions detected from natural scene images. The classifier leverages transfer learning and minimal preprocessing, making it suitable for deployment in resource-constrained environments. To address class imbalance, particularly for the underrepresented Hindi script, the dataset was enhanced through targeted augmentation strategies. The experimental evaluation, including confusion matrix analysis and classification metrics, demonstrated that the model achieves high accuracy with minimal inter-class confusion, even in visually complex multilingual scenarios.

The results confirm that incorporating a script-aware classification module significantly enhances the robustness and adaptability of scene text recognition systems, particularly in the context of Indian languages. As an extension of this work, future research will focus on scaling the classifier to support a broader range of Indian scripts, including Bengali, Gujarati, Tamil, and Malayalam, to accommodate more diverse multilingual scenarios. In addition, the current system is designed for printed text in natural scenes; extending its capabilities to handle handwritten text in scene images remains a promising direction. This would involve training the model on hybrid datasets comprising both printed and handwritten word-level crops, potentially requiring specialized feature extraction layers or transfer learning strategies.

REFERENCES

[1] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” pp. 4510–4520, Jan. 2018, [Online]. Available: http://arxiv.org/abs/1801.04381

[2] A. Kumar Bhunia, A. Konwer, A. Kumar Bhunia, A. Bhowmick, P. P. Roy, and U. Pal, “Script Identification in Natural Scene Image and Video Frame using Attention based Convolutional-LSTM Network.”

[3] V. Naosekpam and N. Sahu, “A Hybrid Scene Text Script Identification Network for Regional Indian Languages,” *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 23, no. 8, Aug. 2024, doi: 10.1145/3649439.

[4] X. Zhou *et al.*, “EAST: An Efficient and Accurate Scene Text Detector,” Apr. 2017, [Online]. Available: http://arxiv.org/abs/1704.03155

[5] S. Ukil, S. Ghosh, S. M. Obaidullah, K. C. Santosh, K. Roy, and N. Das, “Deep learning for word-level handwritten Indic script identification,” Jan. 2018, [Online]. Available: http://arxiv.org/abs/1801.01627

[6] S. Liu, D. Huang, and Y. Wang, “Adaptive NMS: Refining Pedestrian Detection in a Crowd,” Apr. 2019, [Online]. Available: http://arxiv.org/abs/1904.03629

[7] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, “Soft-NMS - Improving Object Detection with One Line of Code,” in *Proceedings of the IEEE International Conference on Computer Vision*, Institute of Electrical and Electronics Engineers Inc., Dec. 2017, pp. 5562–5570. doi: 10.1109/ICCV.2017.593.

[8] T. M. Breuel et al., "High Performance OCR for Printed English and Fraktur Using LSTM Networks," *ICDAR*, 2013.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 6, June 2025

DOI: 10.17148/IJARCCE.2025.14694

[9] Y. Liu, X. Yin, and H. Wang, "Text Detection and Recognition in Natural Scenes," *Pattern Recognition*, 2018.
[10] R. Dhanya, A. G. Ramakrishnan, and P. B. Pati, "Script identification in printed bilingual documents," *Sādhanā*, vol. 27, pp. 73–82, 2002. doi: 10.1007/BF02706427
[11] U. Pal and B. B. Chaudhuri, "Script identification from Indian documents," in *Proc. Int. Conf. Document Analysis and Recognition (ICDAR)*, 1999, pp. 403–406. doi: 10.1109/ICDAR.1999.791805
[12] Tan, M., & Le, Q. V. (2019). "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks." *ICML*.
[13] Kornblith, S., Shlens, J., & Le, Q. V. (2019). "Do Better ImageNet Models Transfer Better?" *CVPR*.
[14] Shorten, C., & Khoshgoftaar, T. M. (2019). "A Survey on Image Data Augmentation for Deep Learning." *Journal of Big Data*.
[15] Sokolova, M., & Lapalme, G. (2009). "A systematic analysis of performance measures for classification tasks." *Information Processing & Management*.