SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V12I9P108

Volume 12 Issue 9, 98-107, September 2025
© 2025 Seventh Sense Research Group®

Original Article

Hybrid Approach for Word Recognition in Bilingual Natural Scene Images

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

[1]*Department of Computer Science, School of Computer Science, Central University of Karnataka. Aand road, Kalaburagi, Karnataka, India.*

[*]*Corresponding Author : gmukarambi@gmail.com*

Received: 08 July 2025 Revised: 10 August 2025 Accepted: 09 September 2025 Published: 29 September 20250

Abstract - *A multilingual country like India, where Kannada and English are the languages derived from the Brahmi and Latin scripts, respectively. In the Indian context, there is a need for Bilingual, Trilingual, and Multilingual word recognition in natural scene images to meet the requirements of a multilingual OCR system. Hence, a novel hybrid-based approach was proposed for extracting the features, such as the Resnet50 architecture in deep learning with 50 layers and utilizing residual learning through skip connections to enable efficient training of very deep networks. A total feature set of size 2048, after implementation of PCA, is reduced to 60 potential features. A dataset of 12,082 real-world sample images is collected, with diverse scenarios where bilingual text appears in various orientations, fonts, and complex backgrounds. In this paper, two experimental setups are carried out: a hybrid-based approach without PCA (Principal Component Analysis) and with PCA. The recognition accuracy was 97.48% using an SVM (Support Vector Machine) classifier without PCA and 97.56% with PCA, respectively. To test the performance of the Resnet50 model, a comparison is made with other pre-trained models like Vgg16, Google Net, Mobile Net, Efficient Net, and Vision Transformer, and later selected an optimum kernel like RBF with an SVM classifier to demonstrate the efficiency of the models. The novelty of this paper is the dimensionality reduction of weak features. The Time complexity of the SVM classifier with training and testing is reduced from 22% to 10% and it also demonstrates the capability of deep learning models to handle the complexities of bilingual text word recognition. It provides an effective solution in a multilingual environment.*

Keywords - *Word Recognition, Bilingual, ResNet50, PCA.*

1. Introduction

In recent years, deep learning has developed in the field of computer vision, enabling breakthroughs in image classification, object detection, and text recognition tasks. Among the numerous deep learning architectures, ResNet-50 has emerged as a widely adopted model due to its ability to address the vanishing gradient problem through residual learning. ResNet-50[8] is a 50-layer convolutional neural network that leverages skip connections to improve feature extraction and training efficiency in complex datasets. Its robust architecture has been pivotal in advancing applications such as bilingual scene text word recognition, where the extraction of meaningful features from multilingual natural scene images is critical. The motivation of this work stems from the growing need to develop efficient and accurate systems for bilingual/ trilingual/ multilingual scene text word recognition, particularly for languages such as Kannada and English. In India, a multilingual country with diverse scripts and languages, robust text recognition systems can facilitate better accessibility in education, transportation, and digital documentation. The inherent challenges of recognizing text in natural scenes, such as variations in font styles, sizes, orientations, and background noise, further necessitate the exploration of advanced models like ResNet-50. Moreover, optimizing these systems for computational efficiency is crucial for real-time bilingual applications in resource-constrained environments. This study investigates the performance of ResNet-50 as part of a bilingual text word recognition system for Kannada and English. By comparing its effectiveness against other classifiers and analyzing the impact of integrating Principal Component Analysis (PCA), the paper aims to enhance both the accuracy and efficiency of the scene text word recognition system. This research provides valuable insights into developing scalable solutions for multilingual text recognition, addressing real-world challenges and promoting technological inclusivity.

The organization paper is divided into four sections. Section 1 presents the introduction. Data collection and analysis are given in Section 2, methodology appears in Section 3, and experimental results and discussion appear in Section 4. Finally, the conclusion is given in Section 5.

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

1.1. Background Study

Albalawi BM. et al. 2024 [1] proposed a novel end-to-end framework for bilingual (Arabic and English) text recognition in natural scene images. The system uses pre-trained CNNs, i.e., ResNet and EfficientNetV2, for text localization with kernel representation to detect multi-oriented and curved text. For recognition, RNN models with an attention mechanism are employed to capture contextual information. The model was evaluated on EvArest, ICDAR2017, and ICDAR2019 datasets. EvArest includes bilingual Arabic-English text, while ICDAR2017 and ICDAR2019 contain multi-oriented and curved text, primarily in English but with some non-Latin scripts. The system outperformed existing methods, the BiLSTM model improved recognition by 3–5% over LSTM, particularly for curved Arabic text, with an F1-score of approximately 0.85 on ICDAR2019. EfficientNetV2 outperformed ResNet in localization by 2–4% in precision.

Alex Noel Joseph Raj, et al. 2022 [2] The system uses Faster R-CNN to extract probable text regions, followed by the rearrangement of text regions into consecutive frames along the time axis. Global and local shape features are extracted using an Extended Histogram of Oriented Gradients (EHOG) from three orthogonal planes. A simple classifier predicts text vs. non-text regions. The MSRA-TD500 dataset, which includes bilingual text (English and Chinese), was used for evaluation. The dataset contains 500 images with multi-oriented text. The framework achieved an F1-score of 0.70, improving detection accuracy by 5–7% compared to traditional methods like MSER-based approaches.

Shi, C.et al. (2013) [3] proposed a part-based tree-structured model for scene text recognition that is adaptable to bilingual settings. It uses a CNN to detect character parts and a tree-structured model to group parts into words. The system employs Connectionist Temporal Classification (CTC) for sequence prediction, which is suitable for multi-script recognition. Evaluated on ICDAR2013 and Street View Text (SVT) datasets, which are predominantly English but include some non-Latin text instances. The model achieved a word recognition accuracy of 88% on ICDAR2013 and 82% on SVT. For bilingual settings, synthetic data augmentation improved performance by 3–4%, but specific bilingual results were not reported. Liu, X. et al. 2016 [4] Proposed method for character recognition in natural scenes, applicable to bilingual settings. (PCA) denoises character images by recovering low-rank components and filtering sparse noise. HOG features are extracted, followed by a sparse representation-based classifier for recognition. The method is tested on English and Chinese characters. Evaluated on different datasets, Char74K, IIIT5K, SVT, and ICDAR2003. Achieved accuracy respectively 67%, 76%, 79% and 75%. Karan Maheshwari et al. 2019 [5] proposed an algorithm to detect texts and non-text in different dialects and orientations of natural scene images. Firstly, features are extracted by probable text regions, and then a combination of statistical filters. Finally, they used text and non-text classifier Artificial Neural Networks (ANN). The proposed method was evaluated on the MSRA-TD500 dataset, and the archived F1 score is 0.67. Veronica et al. 2023 [6] proposed Indian regional language identification, initially created the IIITG-MLRIT2022 dataset, and applied it on the Deep Ensemble Baseline and Ensemble (CNN, ResNet50, DN) models. Achieved F1 scores are 88.40% and 88.04% respectively. Ankan Kumar et al. 2018 [7] proposed a script identification in natural scene images. Firstly, the image was converted into patches, then global and local features were extracted from individual patches. Later, the Softmax function was applied for classification. The proposed method was evaluated on public datasets MLe2e, ICDAR-17, CVSI-15, and SIW-13, which achieved accuracy of 96.70%, 96.50%, 90.23% and 97.75% respectively.

Table 1. Existing literature

Author	Method	Dataset	Results in accuracy (%)
Albalawi BM. et al. 2024 [1]	ResNet with lstm	EvArest,	97.06
		ICDAR2017	98.5
		ICDAR2019	78.3
	EfficientNet02 LSTM	EvArest	98.9
		ICDAR2017	57.9
		ICDAR2019	98.8
Shi, C.et al [3]	CNN- part-based tree-structured model	ICDAR03(FULL)	79.30
		ICDAR03(50)	87.44
		ICDAR11(FULL)	82.87
		ICDAR11(50)	87.04
		SVT	73.5
Liu, X. et al. [4]	HOG-PCA	Char74K	67
		IIIT5K	76
		SVT	79%
		ICDAR2003	75
Veronica et al. [6]	Deep Ensemble(CNN, ResNet50, DN)	IIITG-MLRIT2022	88.40
	Base Ensemble		88.04
Ankan Kumar [7]	LSTM-CNN	MLe2e	96.70
		ICDAR-17	96.50
		CVSI-15	90.23
		SIW-13	97.75

2. Data Collection and Its Analysis

The dataset of bilingual natural scene text (Kannada and English) is not available in the literature. Therefore, a 50-megapixel mobile camera (OPPO Reno10) is used to create the database, where in the Kalaburagi and Bidar districts, a total of 795 samples were captured. Then, we used the CRAFT region-wise segmentation model to separate words from

Kannada and English natural scene text images. After segmentation, a total of 12,082 bilingual scene text words, which include 6,019 Kannada and 6,063 English words. It covers a variety of categories like wall paint, stone, signboard, and iron. The data set contains a number of issues, including low-quality images, complicated backdrops, and perceptual distortion from various lighting situations.

Fig. 1 A sample of original natural scene text image

Fig. 2 Workflow of word segmentation from a natural scene image

2.1. Pre-Processing

To enhance model performance, the images were pre-processed. The dataset was prepared by collecting bilingual scene text images using a smartphone camera and supplementing them with publicly available resources. All images were resized to a fixed resolution of 224 × 224 pixels for compatibility with ResNet50, and converted to grayscale when necessary while retaining RGB channels for CNN input.

To enhance text clarity, pixel intensities were normalized to the range [0,1] for stable training. Candidate word regions were segmented using CRAFT and region-based segmentation methods, after which data augmentation techniques such as rotation, scaling, and flipping were employed to increase variability and improve generalization. Finally, each segmented word image was manually annotated as either Kannada or English, providing accurate labels for supervised training. This detailed pipeline ensures that the preprocessing methodology is transparent, consistent, and fully reproducible.

Fig. 3 Pre-processed sample

3. Proposed Methodology

The proposed architecture combines the strength of deep learning and traditional machine learning for robust image classification. Initially, the input image of size 224×224×3 is passed through a deep convolutional neural network, ResNet-50, which extracts high-level semantic features through its 50-layer residual learning framework. These features, typically 2048-dimensional vectors, capture essential patterns such as shapes, textures, and structures present in the input image.

However, such high-dimensional vectors can be computationally intensive and may introduce redundancy. To address this, Principal Component Analysis (PCA) is applied to reduce the dimensionality of the feature vectors while preserving the most informative components [9].

The resulting reduced feature vector significantly decreases computational complexity and enhances classifier performance. Finally, a Support Vector Machine (SVM) classifier is employed to perform the classification task using the reduced features.

SVM is particularly effective in handling lower-dimensional, linearly or non-linearly separable data, especially when used with kernels such as the Radial Basis Function (RBF).

This integrated pipeline, ResNet-50 for feature extraction, PCA for dimensionality reduction, and SVM for classification, offers a powerful solution for image-based tasks such as scene text recognition or multilingual word classification.

3.1. Feature Extraction: ResNet50 Architecture

The input given to ResNet50 is typically an image of size 224×224×3, where 224 is the height and width of the image, and 3 represents the RGB color channels.

Convolutional and Pooling Layers

3.1.1. Initial Convolutional Layer (conv1)

The first layer applies 64 filters of size 7×7×3 with a stride of 2. The convolution operation is defined as:

$$Y_{x,y,z}^{(1)} \sum_{m=1}^{7} \sum_{n=1}^{7} \sum_{c=1}^{3} {}_{m,n,c,k}^{(1)} {}_{k}^{(i)} {}_{x+m-1,y+n-1,z} \quad (1)$$

Where: The $Y_{x,y,z}^{(1)}$ output is at a position (x, y) in the $z-th$ feature map. $f_{m,n,c,k}^{(1)}$ is the $(m, n, c)-th$ element of the $z-th$ filter. $b_z^{(i)}$ is the bias for the $z-th$ filter.

This is followed by a ReLU activation function:

$$Y_{x,y,z}^{(1)} = \max\left(0, y_{x,y,z}^{(1)}()\right) \quad (2)$$

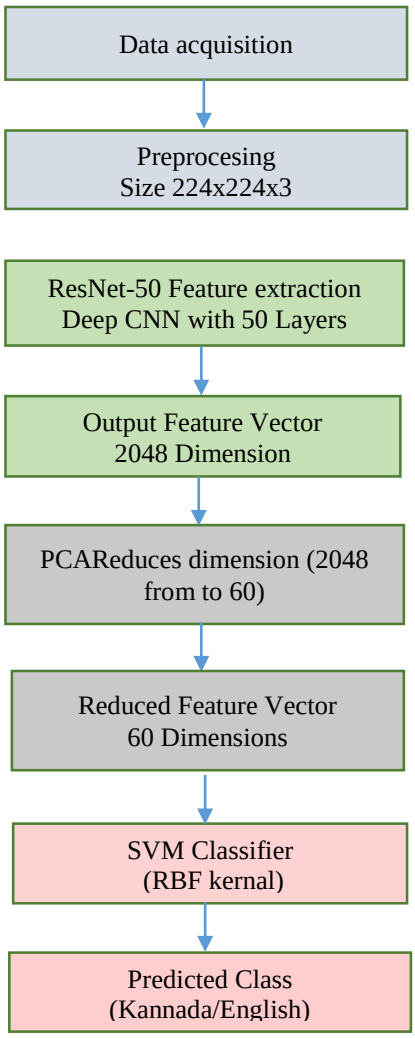

Fig. 4 Proposed methodology

3.1.2. Max-Pooling Layer (pool1)

A max-pooling operation with a 3×3 filter and a stride of 2 is applied to reduce the spatial dimensions:

$$Y_{x,y,z}^{(2)} = max_{m,n} y_{2x+m-1,2y+n-1,z}^{(1)} \quad (3)$$

3.2. Residual Blocks

ResNet50 consists of multiple residual blocks, each containing convolutional layers with skip (residual) connections that add the input to the output of a block.

3.2.1. Residual Block Structure

Each residual block is defined by:

Convolutional Layer: 1×1convolution to reduce dimensions.

$$Y_{x,y,z}^{(3)} = \sum_{c=1}^{d} \quad f_{c,z}^{1*1^{(2)}_{x,y,z_z}{}^{(3)}} \quad (4)$$

Convolutional Layer 2: 3×3 convolution to learn features.

$$Y_{x,y,z}^{(4)} = \sum_{m=1}^{3} \quad \sum_{n=1}^{3} \quad \sum_{c=1}^{d} \quad f_{m,n,c,k}^{3*3} . y_{x+m-1,y+n-1,z}^{(3)_k^{(4)}} \quad (5)$$

Convolutional Layer 3: 1×1 convolution to restore dimensions.

$$Y_{x,y,z}^{(5)} = \sum_{c=1}^{d} \quad f_{c,z}^{(1*1)^{(4)}_{x,y,z_z}{}^{(5)}} \quad (6)$$

Skip Connection: adding to the next convolution layer.

$$Y_{x,y,z}^{(res)} = Y_{x,y,z}^{(5)} + Y_{x,y,z}^{(2)} \quad (7)$$

ReLU Activation: A ReLU function is applied to the sum.

$$Y_{x,y,z}^{(6)} = \max\left(0, y_{x,y,z}^{(res)}\right) \quad (8)$$

3.3. Feature Extraction up to Global Average Pooling Layer(GAP)

After processing through multiple residual blocks, the output feature maps are fed into a global GAP, which computes the average of each feature map.

3.3.1. Global Average Pooling Layer (Avgpool)

Let the input to the average pooling layer be $y_{x,y,z}^{L}$ where *L*is the index of the last convolutional layer. The output of the global average pooling layer is:

$$Y_z^{(avg)} = \frac{1}{H*W} \sum_{x=1}^{H} \quad \sum_{y=1}^{W} \quad y_{x,y,z}^{(L)} \quad (9)$$

Where the height and width of the feature map. $y_z^{avrg}$is the average pooled feature corresponding to the$z-th$ feature map.

This results in a 2048-dimensional feature vector y^{avrg}, which can be used for tasks such as classification

3.4. Principal Component Analysis Layer

3.4.1. Initial Setup

Let the original data matrix be*X*, where *X*has dimensions $NX2048$. Here *N*is the number of samples, and each sample has 2048 features.

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,2048} \\ \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,2048} \end{bmatrix} \quad (10)$$

3.4.2. Mean Centering

First, subtract the mean of each feature from the data to center the data around the origin.

$$X_{centered} = X - \mu \quad (11)$$

Where μ is 1×d, the mean of each column (feature) of X.

3.4.3. Covariance Matrix

Compute the centered data's covariance matrix C. The variance and correlations between the characteristics are captured by the C 2048×2048 covariance matrix.

$$C = \frac{1}{N-1} X_{centered}^{T} X_{centered} \quad (12)$$

3.4.4. Eigen Decomposition:

The eigenvalues and eigenvectors can be found by performing eigen decomposition on the covariance matrix. The primary components are made up of eigenvectors, and the eigenvalues indicate how much variance is captured by each component.

$$Cv_i = \lambda_i v_i \quad (13)$$

Where $v_i$is the$i - th$eigenvector (principal component) and λ_i is the corresponding eigenvalue.

3.4.5. Selecting Principal Components:

Sort the eigenvectors by their corresponding eigen values in descending order. Select the top 60 eigenvectors (principal components) that correspond to the largest eigenvalues. This is what is represented by coeff(:, 1:60) in MATLAB, where coeff contains all the eigenvectors.

Let W_{60}represent the matrix of the top 60 eigenvectors, with dimensions 2048×60.

$$W_{60} = [v_1 v_2 \dots v_{60}] \quad (14)$$

3.5. Project

Project the original 2048-dimensional data onto the 60-dimensional space spanned by these 60 principal components. The reduced data matrix X_reduced will have dimensions N×60.

$$X_{reduced} = X_{centered} W_{60} \quad (15)$$

This matrix multiplication results in each original data point being represented in a 60-dimensional space instead of the original 2048-dimensional space.

3.6. SVM Classifier

Training dataset

$$D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \quad (16)$$

$y_i \in \{-1, +1\}$ ResNet-50 feature vector for the i-th sample (reduced to 60 features)
$x_i \in R^{60}$Corresponding binary class label

3.6.1. RBF SVM: Primal and Decision [10]

$$K(x, x') = exp(-\gamma \parallel x - x' \parallel 2) \quad (17)$$

$$f(x) = sign(w^T x + b) \quad (18)$$

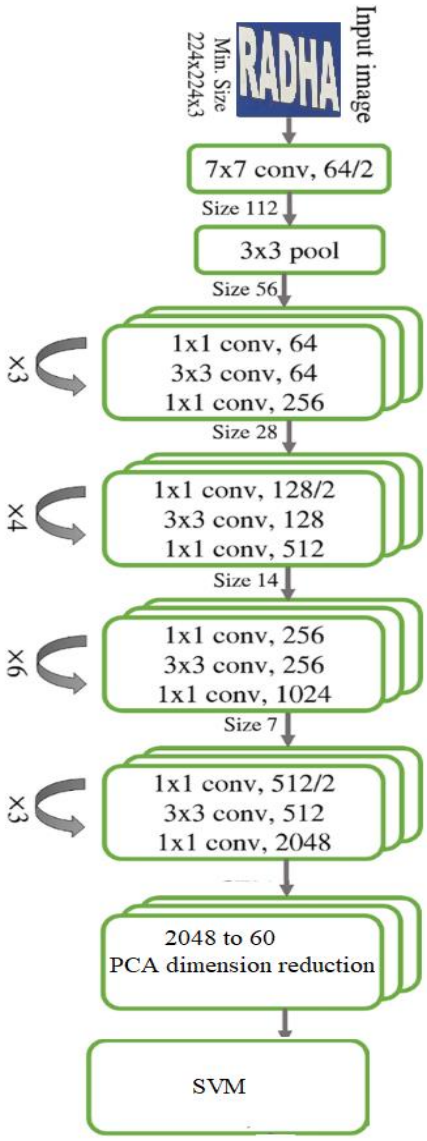

Fig. 5 Proposed ResNet50 architecture

The proposed model utilizes the ResNet50 architecture for feature extraction, followed by Principal Component Analysis (PCA) for feature reduction and classification. ResNet50 employs deep residual learning, allowing efficient training of very deep networks by mitigating the vanishing gradient problem.

Input Layer: Input images were resized to 224 × 224 × 3 to match the ResNet50 input size.

Initial Convolution and Pooling: A 7 × 7 convolution with 64 filters and a stride of 2 was applied, followed by a 3 × 3 max pooling operation with a stride of 2.

Residual Blocks: The network comprises 16 residual blocks organized into four stages with filter depths of 64, 128,

256, and 512. Each residual block follows a bottleneck design (1 × 1, 3 × 3, 1 × 1 convolutions) with batch normalization and ReLU activation. Skip connections were used to enable gradient flow across layers.

Global Average Pooling (GAP): After the final convolutional stage, a global average pooling layer was applied, producing a 2048-dimensional feature vector.

Feature Reduction using PCA: Instead of passing the 2048-dimensional feature vector directly into the classifier, PCA was applied to reduce the feature space to 60 principal components. This significantly decreased computational complexity while preserving the most discriminative variance in the data.

Classification Layer: The reduced 60-dimensional feature vectors were passed through a fully connected layer with softmax activation to predict the language class (Kannada or English)

4. Results and Discussion

The author employed a cross-validation technique with splits of 50 x 50,60 x 40,70 x 30, and 80 x 20 while training the dataset with a feature set of 60.

The observation from Table 1 is that the performance of five classifiers, i.e SVM, KNN, Decision Tree, Random Forest, and Deep Neural Network (DNN), on bilingual (English and Kannada) datasets using a split approach (50:50, 60:40, 70:30).

Table 2. The cross-validation performance of ResNet50 with PCA

Dataset	Classifier	Accuracy	Precision		Recall	
			English	Kannada	English	Kannada
50:50	SVM	0.95516	0.9477	0.9628	0.9631	0.9472
	KNN	0.94954	0.9340	0.9652	0.9652	0.9345
	Decision Tree	0.88600	0.9007	0.8709	0.8775	0.8952
	Random Forest	0.94705	0.9366	0.9578	0.9579	0.9364
	DNeural Network	0.96029	0.9664	0.9541	0.9557	0.9651
60:40	SVM	0.95761	0.9571	0.9581	0.9591	0.9561
	KNN	0.95471	0.9384	0.9715	0.9713	0.9388
	Decision Tree	0.89227	0.8992	0.8852	0.8894	0.8953
	Random Forest	0.95389	0.9416	0.9665	0.9665	0.9416
	DNeural Network	0.97084	0.9837	0.9577	0.9598	0.9828
70:30	SVM	0.96112	0.9582	0.9640	0.9641	0.9582
	KNN	0.95506	0.9434	0.9668	0.9662	0.9443
	Decision Tree	0.89688	0.8978	0.8960	0.8968	0.8970
	Random Forest	0.95506	0.9451	0.9651	0.9647	0.9458
	DNeural Network	0.97298	0.9742	0.9718	0.9720	0.9739

The Deep Neural Network consistently achieved the highest word recognition accuracy, precision, and recall across all splits, with the best result of 97.3% accuracy with a 70:30 split as compared to SVM, KNN, decision tree and Random Forest.

Table 3. ResNet 50 with PCA for optimum cross-validation 80:20

Classifier	Accuracy	Precision		Recall		F1- score		Training time in Sec	Testing time in Sec
		English	Kannada	English	Kannada	English			
SVM	0.9603	0.9630	0.9574	0.9599	0.9607	0.9614	0.9590	165.9873	0.0423
K-NN	0.9665	0.9589	0.9745	0.9754	0.9574	0.9671	0.9659	0.0912	0.0729
DT	0.8862	0.8913	0.8809	0.8877	0.8846	0.8895	0.8828	0.1504	0.0285
RF	0.9686	0.9654	0.9719	0.9732	0.9637	0.9693	0.9677	3.3334	0.2456

Table 4. ResNet 50 without PCA for optimum cross-validation 80:20

Classifier	Accuracy	Precision		Recall		F1- score		Training time in Sec	Testing time in Sec
		English	Kannada	English	Kannada	English	Kannada		
SVM	0.96442	0.9640	0.9649	0.9671	0.9616	0.9655	0.9632	10.566	0.023869
K-NN	0.96111	0.9504	0.9726	0.9737	0.9482	0.9620	0.9604	0.039805	0.71745
DT	0.87878	0.8751	0.8827	0.8886	0.8686	0.8818	0.8756	1.7585	0.0026541
RF	0.96152	0.9528	0.9709	0.9722	0.9505	0.9625	0.9606	12.638	0.2408

The observation of Tables 3 and 4 shows that the Deep Neural Network (DNN) without PCA again delivers the highest recognition accuracy as 97.48%, along with the best precision, recall, and F1-score for both English and Kannada. However, it requires the longest training time (24.04 sec) and a moderate testing time (0.1095 sec). Random Forest (RF) and SVM follow with 96.15% and 96.44% accuracies, respectively. While RF maintains strong precision and recall, its testing time is higher (0.2408 sec) compared to SVM (0.0239 sec), which is efficient in prediction despite a slightly longer training time than RF. K-NN, although achieving good accuracy (96.11%), shows a high testing time (0.717 sec), which is expected due to its instance-based nature. Decision Tree (DT) again performs the weakest with 87.87% accuracy but remains the fastest in prediction (0.0027 sec). In comparison to PCA-based results, these values indicate that PCA helps in reducing training time and improving testing efficiency, especially for SVM and DNN, while maintaining similar or slightly improved accuracy. Without PCA, the models train to require more training time and show marginal differences in performance, but DNN remains the best classifier in both scenarios.

Table 5. Comparative analysis of the proposed method vs existing methods

Models	Features	Accuracy in %
ResNet50	2048	97.56
Vgg16	4096	86.84
Google net	1024	95.76
MobileNet	1280	95.79
Efficient Net	1280	96.50
Proposed modified ResNet50	60	97.48

In binary classification using PCA-reduced ResNet-60 features, the RBF kernel is preferred due to its ability to model complex, non-linear boundaries between two classes. The linear kernel is faster and suitable when classes are linearly separable, but often underperforms in complex visual tasks. The polynomial kernel can capture moderate non-linearity but may overfit on small datasets. The sigmoid kernel is less stable and rarely used. Overall, the RBF kernel offers the best balance between accuracy and generalization for binary image classification problems.

4.1. Comparative Analysis

The comparative analysis in Table 5 highlights the effectiveness of the proposed method against existing deep learning models like ResNet50, Vgg16, GoogleNet, MobileNet, and EfficientNet under the same conditions. The existing ResNet50 with its original 2048-dimensional feature set obtained the accuracy of 97.56% without PCA. The proposed modified ResNet50 method, which uses only 60 optimized features, obtained an accuracy of 97.48% with PCA. This demonstrates the strength of the proposed method with a dimensionality reduction strategy, achieving nearly the same classification performance with significantly lower computational complexity. Rather than other models such as GoogleNet (95.76%), MobileNet (95.79%), and EfficientNet (96.50%), which also perform well, but with higher feature extractor dimensions compared to the proposed method. vgg16, despite having the largest feature vector size of 4096, achieves the lowest accuracy at 86.84%, indicating that larger feature sets do not necessarily translate to better performance.

Table 6. Comparative analysis of a multilingual public dataset

Model	Dataset	Type	Size	Accuracy in %
Our proposed model	CVSI-15	11Language	10688	97.87
	MLe2e	4 Language	1821	89.98
	SIW-13	13 Language	13045	80.40
	Our dataset	2 Language	12082	97.48

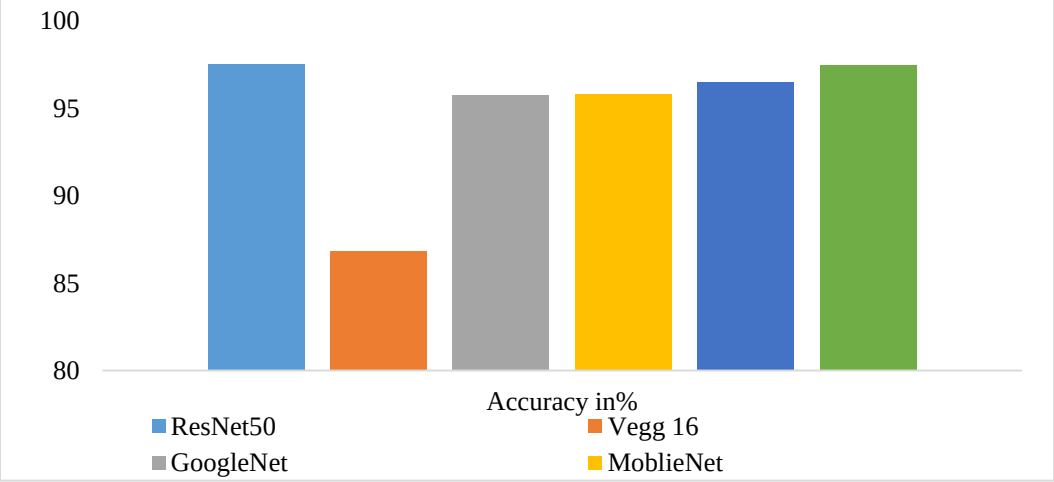

Fig. 6 Graphical representation of Accuracy vs deep learning model

Table 6 Curiosity of testing the strength of the proposed method on multilingual languages, the comparative analysis is carried out on public datasets as follows. The multilingual datasets are like CVSI-15, Mle2e, and SIW-13. It achieves 97.87% accuracy on CVSI-15 (11 languages, 10,688 samples), 89.98% on MLe2e (4 languages, 1,821 samples), and 80.40% on SIW-13 (13 languages, 13,045 samples). These results highlight the model's robustness and effectiveness in handling diverse and complex multilingual scene text images.

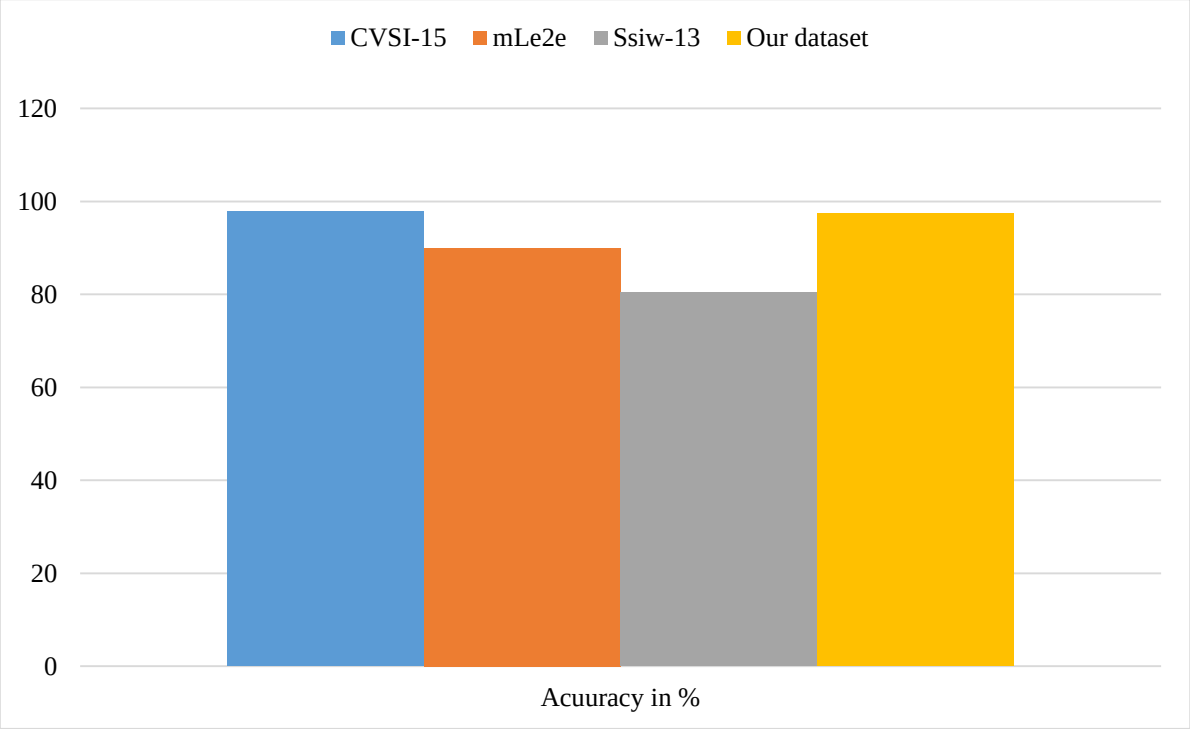

Fig. 7 Graphical representation of performance evaluation on the public dataset

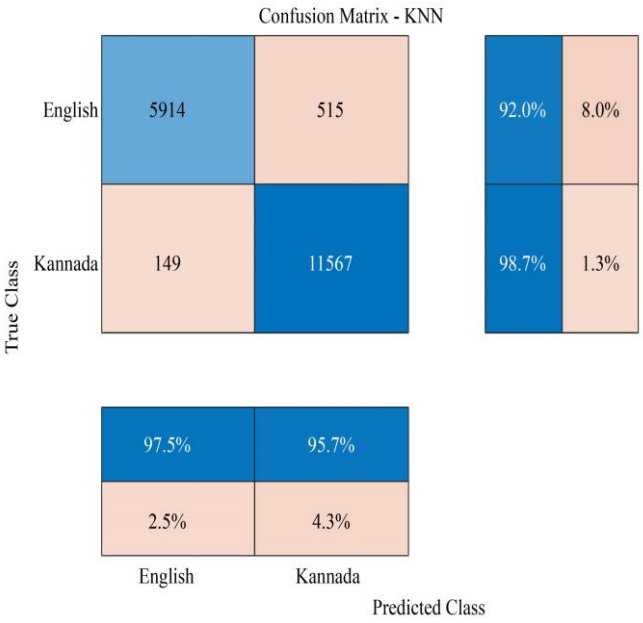

Fig. 8 Confusion matrix of KNN

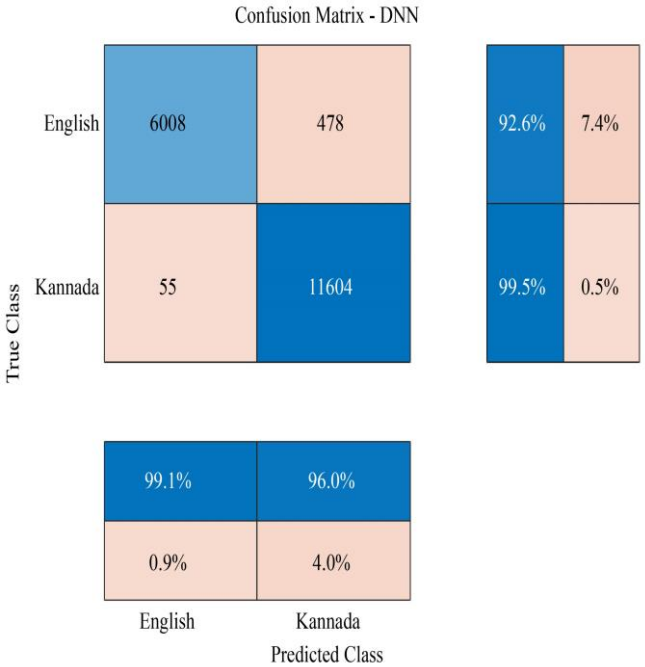

Fig. 9 Confusion matrix of PCA with ResNet 50

The proposed method achieved 97.87% accuracy on the CVSI-15.

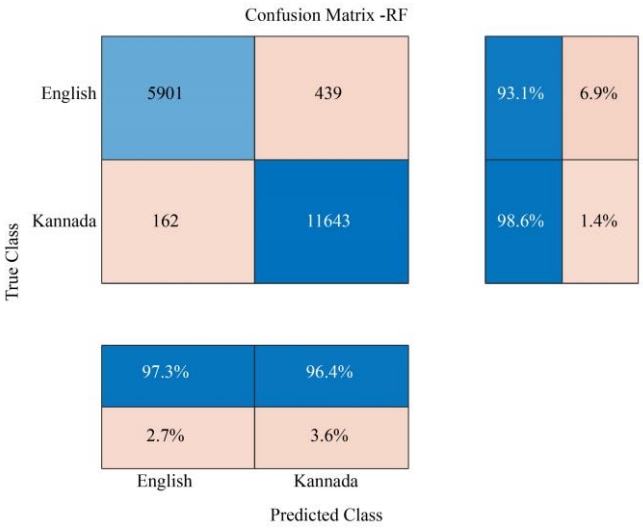

Fig. 10 Confusion matrix of PCA with random forest

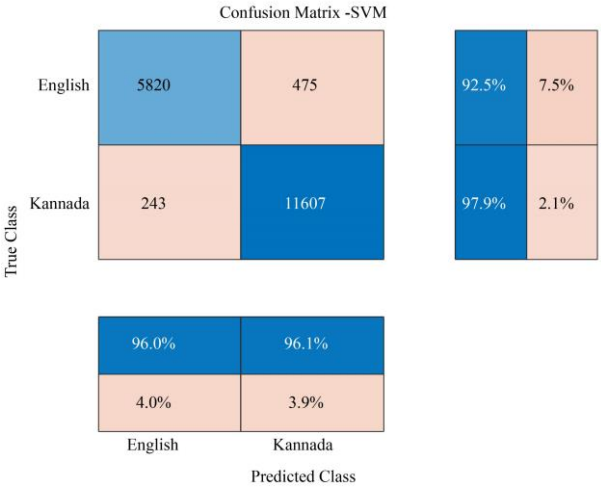

Fig. 11 Confusion matrix of PCA with SVM

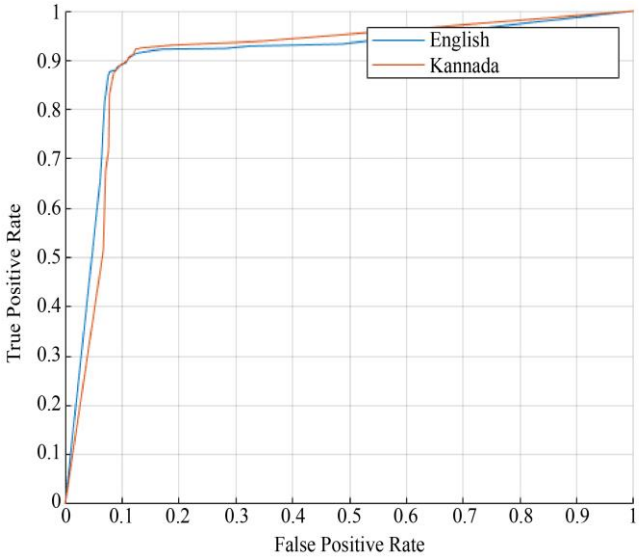

Fig. 12 ROC curve for a decision tree

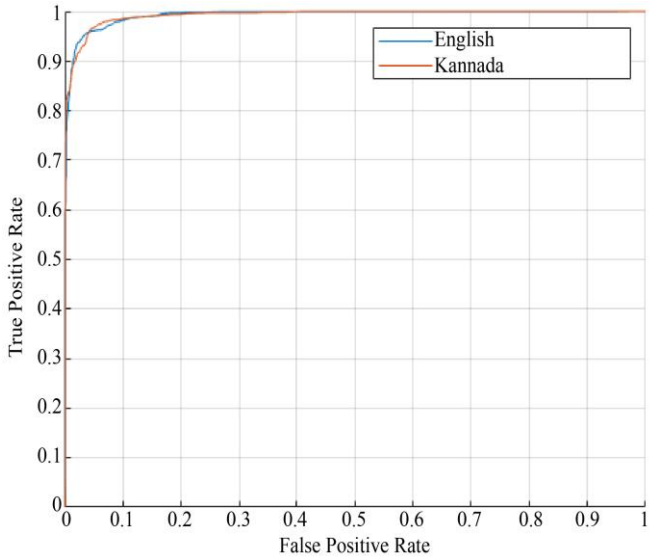

Fig. 13 ROC curve for random forest

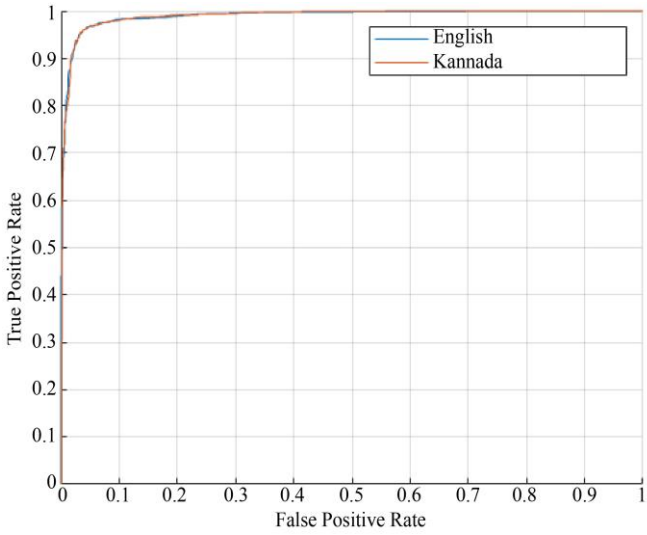

Fig. 14 ROC curve for SVM

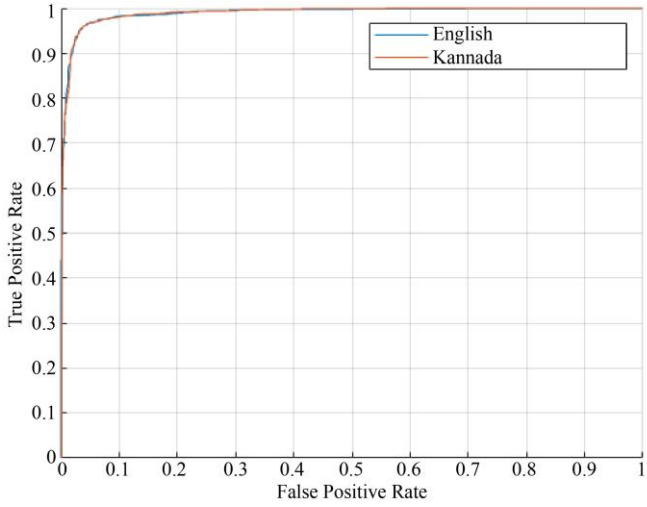

Fig. 15 ROC curve for hybrid approach ResNet50-PCA

4.2. Performance

Precision: The ratio of accurate positive forecasts to all positive forecasts.

$$Precesion = \frac{TruePositive}{TruePositive+FlasePositive} \quad (19)$$

Recall: the proportion of actual positive cases that fulfilled accurate positive projections.

$$Re\,c\,all = \frac{TruePositive}{TruePositive+FlaseNegative} \quad (20)$$

F1 score: The harmonic mean of precision and recall, balancing both metrics.

$$F1Score = 2 * \frac{Pr\,ecision*Re\,call}{Precision+Recall} \quad (21)$$

Accuracy: The proportion of correct predictions (both true positives and true negatives) among all predictions.

$$Accuracy = \frac{TrueNegative+TruePositive}{TrueNegative+FlasePositive+TruePositive+Flasenegative} \quad (22)$$

4.3. Experimental Setup

The experimental setup was implemented using MATLAB R2023b with the Deep Learning Toolbox, running on a system equipped with a GPU Quadro RTX 5000, with 17GB of memory and 48 multiprocessors. 256 GB RAM. The ResNet50-based model was trained with a constant learning rate of 0.001 using the Stochastic Gradient Descent with Momentum (SGDM) optimizer. A batch size of 32 and 30 training epochs were employed, with validation performed every 50 iterations. Cross-entropy loss was used as the objective function, and L2 regularization with a weight decay of 0.0001 was applied to minimize overfitting.

5. Conclusion

In this paper, a novel hybrid approach, like ResNet50 with PCA feature extraction, is proposed for Bilingual Kannada and English scene text word recognition in natural scene images. The proposed method demonstrated its effectiveness on a dataset of 12,082 real-world images. Obtained an accuracy of 97.48% for the proposed ResNet50. The proposed method improved efficiency by significantly reducing training and testing time and offered a balanced recognition performance across both languages. It also carried out the comparison with a hybrid approach, ResNet50 with PCA and without PCA. The Hybrid ResNet50 with PCA provided an overall better trade-off between accuracy, computational efficiency, and generalization. The results highlight the capability of the proposed method to address the complexities of bilingual text recognition in multilingual environments, paving the way for scalable solutions in real-world scenarios. The future research is expanding to Indian Multilingual languages with modified Deep Learning methods to get the highest accuracy with a smaller number of features.

Funding Statement

The Authors are grateful to KATePS. Govt of Karnataka, India, for support in carrying out the research work. (DST/KSTePS/Ph.D. Fellowship/MP-12:2023-24/350/2)

Referancess

[1] Bayan M. Albalawi et al., "An End-to-End Scene Text Recognition for Bilingual Text," *Big Data and Cognitive Computing*, vol. 8, no. 9, pp. 1-40, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[2] Alex Noel Joseph Raj et al., "Bilingual Text Detection from Natural Scene Images using Faster R-CNN and Extended Histogram of Oriented Gradients," *Pattern Analysis and Applications*, vol. 25, pp. 1001-1013, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[3] Cunzhao Shi et al., "Scene Text Recognition using Part-Based Tree-Structured Character Detection," *2013 IEEE Conference on Computer Vision and Pattern Recognition*, Portland, OR, USA, pp. 2961-2968, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[4] Zheng Zhang, Yong Xu, and Cheng-Lin Liu, "Natural Scene Character Recognition using Robust PCA and Sparse Representation," *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, Santorini, Greece, pp. 340-345, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[5] Karan Maheshwari et al., "Bilingual Text Detection in Natural Scene Images using Invariant Moments," *Journal of Intelligent & Fuzzy Systems*, vol. 37, no. 5, pp. 6773-6784, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[6] Veronica Naosekpam, and Nilkanta Sahu, *Multi-Label Indian Scene Text Language Identification: Benchmark Dataset and Deep Ensemble Baseline*, 1st ed., Intelligent Systems and Applications in Computer Vision, CRC Press, pp. 1-21, 2023. [Google Scholar] [Publisher Link]

[7] Ankan Kumar Bhunia et al., "Script Identification in Natural Scene Image and Video Frames using an Attention based Convolutional-LSTM Network," *Pattern Recognition*, vol. 85, pp. 172-184, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[8] Ashwaq Khalil et al., "Text Detection and Script Identification in Natural Scene Images using Deep Learning," *Computers & Electrical Engineering*, vol. 91, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[9] Jingsi Zhang et al., "A Novel ResNet50-based Attention Mechanism for Image Classification," *Journal of Applied Science and Engineering*, vol. 27, no. 8, pp. 2961-2969, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[10] Ji Ma, and Yuyu Yuan, "Dimension Reduction of Image Deep Feature using PCA," *Journal of Visual Communication and Image Representation*, vol. 63, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[11] Karl Thurnhofer-Hemsi et al., "Radial basis Function Kernel Optimization for Support Vector Machine Classifiers," *arXiv Preprint*, pp. 1-17, 2020. [CrossRef] [Google Scholar] [Publisher Link]