Journal of Intelligent Systems and Internet of Things
Vol. 14, No. 02, PP. 260-273, 2025
DOI: https://doi.org/10.54216/JISIoT.140220

Dynamic Feature Weighting for Efficient Multi-Script Identification Using YafNet: A Deep CNN Approach

Yahia Menassel[1,*], Rashiq Rafiq Marie[2], Faycel Abbas[1], Abdeljalil Gattal[1], Mohammed Al-Sarem[3]

[1]Laboratory of Vision and Artificial Intelligence (LAVIA), Echahid Cheikh Larbi Tebessi University, Tebessa, Algeria
[2]College of Computer Science and Engineering, Taibah University, Madinah, Saudi Arabia
[3] Department of Information Technology, Aylol University College, Yarim, Yemen

Emails: yahia.menassel@univ-tebessa.dz; rmarie@taibahu.edu.sa; faycel.abbas@univ-tebessa.dz; abdeljalil.gattal@univ-tebessa.dz; mohsarem@gmail.com

Abstract

Script identification is crucial for document analysis and optical character recognition (OCR). This study proposes YafNet, a novel convolutional neural network (CNN) architecture, developed from scratch, to tackle the challenges of script identification in both handwritten and printed word images. YafNet dynamically weights features, enabling it to learn and combine multimodal features for accurate script identification. To evaluate its efficacy, we use the imbalanced ICDAR 2021 Script Identification in the Wild (SIW 2021) competition dataset. Experimental results demonstrate that YafNet outperforms conventional approaches, particularly when trained on mixed handwritten and printed data. It achieves high classification accuracy, balanced accuracy, and ROC AUC scores, indicating its robustness and generalizability. The incorporation of data augmentation and external data further enhances performance, underscoring the model's potential for real-world applications.

Received: May 01, 2024 Revised: July 25, 2024 Accepted: November 18, 2024

Keywords: Script identification; YafNet; CNN; Imbalanced dataset

1. Introduction

Text interpretation consists of three main challenges: text detection, recognition, and script identification [1]. Script identification is essential in Optical Character Recognition (OCR), and it has become increasingly important as multimedia data has grown in volume. While previous methods have mostly focused on script identification in videos, our study focuses on script identification in document images which is a relatively unexplored field. This task is particularly challenging because of variations in text appearance, image quality, diverse text styles, complex backgrounds, and subtle script differences. Furthermore, script identification gets more difficult when certain scripts have common characters.

Compared to the broader fields of document analysis and optical character recognition, research on script identification remains limited. The majority of previous studies has been focused on recognizing major scripts, such as English, Arabic, Chinese, and Indian, with little consideration for other scripts. Furthermore, there is an increasing need for research into script identification in printed and handwritten documents, as well as video and camera-based images, especially as mobile and low-cost devices become more popular [2-3].

This paper presents a comprehensive overview of several script identification, which are discussed in the relevant sections. We review both traditional handcrafted machine learning techniques [4-6] and deep learning methods [7-10]. Handcrafted methods focus on extracting features from script images, such as textures, edges, and contours, which are required for classification. Traditional texture analysis techniques, such as Local Binary Patterns (LBPs) and their variants [17-18], compactly encode texture information. Edge detection techniques identify character edges and contours, which provide useful shape information for classification. Techniques such as the texture feature column scheme [19] provide a statistical depiction of scripts through analysing pixel intensity distributions.

Combining classical image processing with deep learning models improves feature extraction. Several studies have combined the interpretability of traditional image processing methods with deep learning models' powerful feature extraction capabilities [2-4]. For example, prior to classification, CNN-extracted features can be enhanced using techniques such as edge or texture detection, leading to improved performance.

Deep learning techniques [11-13], particularly Convolutional Neural Networks (CNNs), have revolutionized script identification by automating feature extraction and enabling the fusion of multimodal features such as visual, structural, linguistic, and metadata cues into unified models. These networks capture intricate patterns across diverse data dimensions, significantly improving classification accuracy, especially in complex scenarios like low-quality scans or scripts with subtle differences. This approach allows for a deeper understanding of the scripts' characteristics and enhances the model's ability to classify printed and handwritten documents more effectively.

Deep CNNs [14-16] are becoming popular due to their capacity to learn and extract hierarchical features from raw script input. These models, which are made up of layers such as convolutional, pooling, and fully connected layers, can detect complex patterns in different scripts. More advanced architectures include regularization methods like dropout, training stabilization techniques like batch normalization, and vanishing gradient mitigation methods like residual connections. The key benefits of creating a CNN from scratch for script identification include end-to-end training tailored to the task and ensuring feature extraction directly from input data without the use of other tools. Additionally, we utilize data augmentation techniques to improve the model's performance in script identification. This strategy emphasizes the model's ability to withstand class imbalance and provides a solid baseline without the use of pretrained models, ensemble models, or post-processing. In addition, it highlights the adaptability and scalability of a purely data-driven CNN architecture for further studies and applications.

The rest of this paper is structured as follows: Section 2 summarizes the extant literature on script identification techniques. Section 3 describes the proposed methodology for identifying scripts in handwritten and printed word images using YafNet, a custom CNN architecture developed from scratch. Section 4 presents the benchmarking strategy and experimental results, comparing YafNet's performance to other methods across a variety of evaluation metrics. Finally, Section 5 concludes the study, summarizing the important findings and discussing potential future research directions.

2. Related Work

Script identification is a well-documented issue within the document image analysis community. [1] And [2] provide thorough overviews of various methods for addressing this challenge. These methods are commonly used on two sorts of datasets: handwritten or printed document images and text in scene images.

Singh et al. (2016) [5] proposed a method for automatically recognizing the script of text in scene images. The technique represents text images using mid-level features derived from intensively computed local features. An off-the-shelf classifier is subsequently employed for script identification. The technique requires minimal labelled data and achieves a 96.70% accuracy on the CVSI dataset. An additional Indian Language Scene Text (ILST) dataset is provided as well for more challenging evaluation, which includes text in six Indian scripts. The method outperforms traditional texture-based and some deep learning methods while being robust and computationally efficient. Shi et al., (2016) [7] presented a method for identifying scripts in natural images using a deep learning model called Discriminative Convolutional Neural Network (DiscCNN). To successfully differentiate scripts, the technique combines deep features with mid-level representations. DiscCNN is optimized to handle various challenges, including text variances, quality issues, and subtle script differences. The method also introduces the SIW-13 dataset, which has 16,291 images distributed among 13 scripts. Experimental results from multiple datasets demonstrate the superiority of DiscCNN in script identification tasks.

Bhunia et al. (2019) [8] developed a novel attention-based Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) network for identifying scripts in natural scene images and video frames. This approach aims to overcome challenges such as poor image quality, complex backgrounds, and script similarity (e.g., Greek and Latin). The effectiveness of this approach was evaluated on four publicly available datasets (SIW-13, CVSI-2015, ICDAR-17, and MLe2e), and it was found to outperform existing methods. Previous studies often used convolutional neural networks (CNNs) for script identification or fully convolutional networks (FCNs) for model enhancement but not classification. Khalil et al. (2021) [9] proposed two end-to-end approaches, multi-channel masking (MCM) and multi-channel segmentation (MCS), for training text recognition and script identification models simultaneously.

Zhang et al. (2023) presented two approaches to multilingual scene text understanding [10, 20]. The first is EA-ConvNext [10], an improved script identification method based on ConvNeXt. Key improvements include creating edge flow maps to increase script count while reducing noise, as well as introducing a coordinate attention module to enhance vertical spatial feature description. The SIW-13 dataset is expanded to include Uyghur script images, forming SIW-14. Evaluating the approach using public and extended datasets to confirm its efficacy. The second

[20] uses a residual network to extract shallow and semantic features, which were then merged with global features extracted using the swin transformer. Li et al. (2023) [11] proposed a Self-Attention Network (SANet-SI) using a multi-scale feature extraction approach to collect both local and global features. SANet-SI uses a Style-based Recalibration Module (SRM) to highlight dominant features and Global Average Pooling (GAP) instead of Fully Connected layers to increase efficiency.

Table 1: performance comparison of well-known script identification methods

Study	Approach	Dataset	Accuracy
Pati et al. (2008) [4]	Gabor using SVM classifiers	Pati dataset	94.80%
Singh et al., (2016) [5]	Mid-level feature and an off-the-shelf classifier	ILST CVSI2015	88.67% 99.09%
Shi et al., (2016) [7]	Discriminative Convolutional Neural Network (DiscCNN)	SIW-13 CVSI2015 Pati dataset	88.70% 96.10% 99.30%
Bhunia et al. (2019) [8]	CNN-LSTM framework	SIW-13 CVSI2015 ICDAR-17 MLe2e	96.50% 97.75% 90.23% 96.70%
Khalil et al. (2021) [9]	The end-to-end trainable MCM and MCS	ICDAR-17 MLe2e	54.34% 81.13%
Das et al. (2021) [3]	HRnet48, OCRnet and DCN Resnet-101, Resnest-200 and Densenet-121 ResNet50 and NFnet-f3 ResNet and EfficientNet EfficientNet Dense Multi-Block Linear Binary Pattern Oriented Basic Image Feature columns	SIW 2021	**99.84%** 98.87% 97.17% 97.09% 94.79% 94.45% 83.83%
Zhang et al. (2023) [10]	ConvNeXt, Edge Flow, and Coordinate Attention	CVSI-2015 MLe2e SIW-13 SIW-14	97.30% 93.50% 92.40% 92.00%
Zhang et al. (2023) [20]	Res2Net50, Feature Pyramid Network (FPN), Adaptive Spatial Feature Fusion (ASFF) and Swin transformer	CVSI-2015 SIW-13	96.00% 94.70%
Li et al. (2023) [11]	Self-Attention Network (SANet-SI)	RRC-MLT2017 SIW-13 CVSI2015	95.33% 96.18% 99.03%

Peng et al. (2024) [12]	End-to-end CNN, Squeeze and Excitation (SE) and LSTM	MLe2e ICDAR-17 SIW-13 CVSI-2015	97.66% 90.24% 96.66% 98.44%
Ferrer et al. (2024) [14]	ResNet Model VGG Model	MDIW-13: Printed words MDIW-13: HW words MDIW-13: Printed words MDIW-13: HW words	96.46% 87.72% 94.06% 77.69%
Jindal (2024) [15]	Convolutional and recurrent layers	MDIW-13 PHDIndic_11	97.57% 96.15%

To address script identification in handwritten and printed document images, Pati and Ramakrishnan (2008) [4] proposed method effectively identifies scripts at the word level, providing a robust solution for multi-script document processing in OCR systems. Their method evaluates Gabor and discrete cosine transform (DCT) features separately before combining them using a weighted majority-voting scheme. Combining Gabor features with nearest neighbour or SVM classifiers yields promising results on a large dataset of 220,000 scanned word images from 11 different scripts (Pati dataset).

Das et al. (2021) [3] provided a summary of the 1st Competition on Script Identification in the Wild (SIW 2021), which was held in conjunction with the 16th International Conference on Document Analysis and Recognition. The competition aims to enhance script identification techniques using a large-scale dataset of 13 scripts (MDIW-13 dataset) from handwritten and printed word scenarios. The competition featured three distinct tasks, each based on the type of training and testing data. Out of the 19 registered research groups, six teams from academia and industry advanced to the final round, presenting 166 algorithms for evaluation. The contributions included both deep learning models and traditional image processing methods. The results revealed that deep learning methods outperformed conventional statistical techniques, with the top-performing model achieving an impressive 99% classification accuracy across all three tasks, which included over 50,000 test samples. However, the results also indicated areas for improvement, particularly in handwritten samples and specific scripts.

Peng et al. (2024) [12] introduced an end-to-end CNN with dual streams for script identification in document images, which uses enhanced Squeeze-and-Excitation (SE) for visual features and LSTM for spatial dependencies. An adaptive fusion technique combines streams' outputs with weights learned during training. On four public datasets, the method achieves high accuracies, outperforming state-of-the-art results. Ablation experiments confirm the effectiveness of each component. Gupta et al. (2024) [13] proposed using MobileNetV3 to identify scripts in Indian language document images, hence improving OCR efficiency. MobileNetV3, a CPU-friendly CNN, is used to develop an accurate, fast, and efficient script identification module. The study focuses on identifying six Indian scripts and English using four different experimental setups. This methodology addresses real-world challenges and enhances robustness in OCR tasks. Ferrer et al. [14] (2024) developed the MDIW-13 database for script identification, which contains 13 scripts and more than 87,000 handwritten and printed words. This database is particularly helpful in Indic contexts due to its variety of scripts, many of which are widely used in real-world applications in India. The authors further note that the database was carefully pre-processed to remove background noise, ensuring data accuracy. In addition, the study provides a benchmark for script identification by evaluating performance in terms of words, lines, or pages. The results show that the ResNet surpasses VGG in terms of accuracy for both printed and handwritten data. Jindal (2024) [15] proposed a deep learning method that combines convolutional and recurrent layers to identify scripts in multi-script handwritten and printed words. This approach extracts deep hierarchical and temporal features while resolving issues like character similarity and noise. Experiments on the MDIW-13 and PHDIndic_11 datasets reveal better performance over existing approaches, allowing OCR systems to digitize text across multiple scripts. Table 1 summarizes significant contributions to handwritten or printed document images, as well as text in scene images from literature.

3. Methodology

To implement our convolutional neural network (YafNet) from scratch for script identification, we follow a detailed systematic process organized into distinct phases, ensuring both clarity and effectiveness throughout.

- *Preprocessing*: The first step involves converting the images into a format suitable for CNN processing. This typically includes resizing the images to a uniform size, ensuring consistency in the input dimensions. Additionally, the pixel values are normalized to a range between 0 and 1 by scaling them from their original range (0-255). This normalization helps the network converge faster and improves stability during training.
- *Define the CNN architecture*: YafNet consists of numerous layers built together to extract and learn features from input images. The architecture begins with convolutional layers that apply convolution operations to detect features such edges, textures, and patterns. These layers use filters to slide across the input data, producing feature maps that highlight specific images characteristics. Next, batch normalization layers are applied to the convolutional layer output, which speeds up training while also adding a regularization effect to prevent overfitting. The rectified linear unit (ReLU) activation function is used to add nonlinearity, allowing the model to learn complex patterns. MaxPooling Layers then reduce the spatial dimensions of the feature maps, leaving just the most important features while reducing computational costs and the risk of overfitting. A flatten layer is applied to transform 2D feature maps into 1D vectors, which are then fed into fully connected (dense) layers for final classification. The last dense layer uses a softmax activation function to generate probabilities for each class, which represent the model's confidence in predicting the correct script.
- *Compile and train the model*: In this step, we compile the model via Adam optimizer, which adjusts the learning rate during training to accelerate convergence and improve performance. Here, parameters such as batch size (how many samples are processed before updating the model's weights) and epochs (how many times the model receives the whole dataset during training) are determined. These steps ensure that the CNN is effective in script identification, yielding high accuracy and robust performance.

Figure 1 presents the proposed end-to-end training from scratch approach (YafNet). Developing and training the CNN model from scratch allows a fully customized architecture specifically adapted for the script identification task. This method eliminates the dependence on pretrained models, ensuring that the model learns all features directly from the provided data. Without using external detection or alignment techniques, the CNN learns the key features required for script identification from the input images. This shows the model's ability to automatically detect and process relevant features in a purely data-driven manner. Furthermore, using a single CNN model—rather than an ensemble—simplifies both the architecture and the training process, demonstrating the model's ability to tackle script identification complexity without relying on the combined strengths of multiple models. In addition, the proposed approach intentionally avoids using post-processing techniques to improve classification results. This demonstrates the raw performance and resilience of the CNN architecture, as the results reflect the model's direct learning process with no external adjustments.

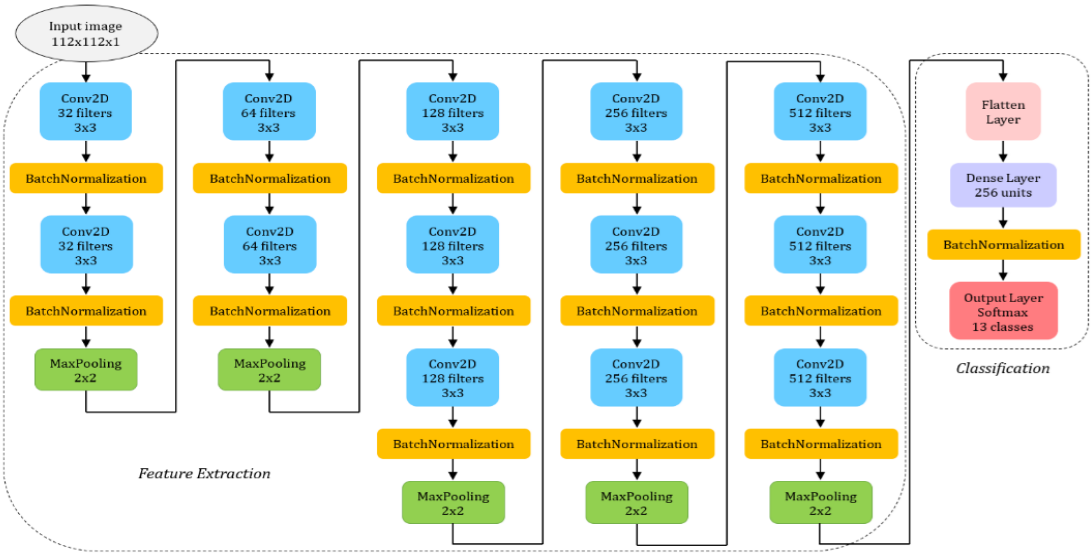

Figure 1. Proposed YafNet Architecture

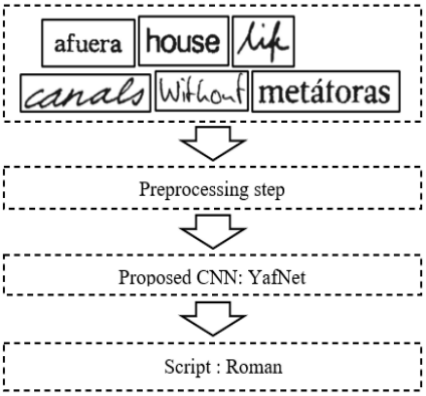

Figure 2. Steps of the proposed method.

By focusing on these contributions, the development of YafNet—a CNN built from scratch for script identification—underscores the significance of pure, data-driven learning. It highlights the inherent ability of neural networks to independently address complex classification tasks without the aid of external features or pretrained models. Figure 2 presents an overview of the proposed method.

A. Dataset

Several publicly accessible datasets contain images of handwritten and printed documents, including the Pati dataset [4], MDIW-13 dataset [14], PHDIndic_11 [21], and the SIW 2021 Competition [3]. SIW 2021 remains to be one of the major publicly available multiscript datasets for handwritten and printed document images in terms of script count, size, and availability. Additionally, the PHDIndic_11 dataset, which focuses on 11 official Indic scripts, and the MDIW-13 dataset by Ferrer et al. (2024), which includes 13 scripts and over 87,000 handwritten and printed words, have improved the field of script identification.

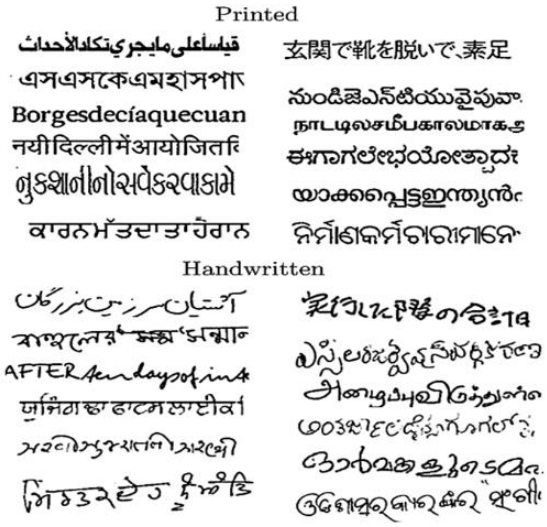

Figure 3. Example of image word samples used in the competition

In this study, we used the SIW 2021 Competition dataset [3] to evaluate our script identification method. This dataset includes a significant number of words extracted from both handwritten and printed documents. Figure 3 displays a set of word images in 13 different scripts: Arabic, Bengali, Gujarati, Gurmukhi, Devanagari, Japanese, Kannada, Malayalam, Oriya, Roman, Tamil, Telugu, and Thai. During the competition, the dataset was split into training and testing sets, as shown in Table 2. The training data were subsequently divided into two categories: printed images (21,974) and handwritten images (8,887). The testing data included 55,814 unlabeled images, comprising both handwritten and printed samples, identified using 13 distinct scripts.

Table 2: overview of the word images used in each script for the training and testing sets

Script	Training		Testing	
	Printed	*Handwritten*	*Printed*	*Handwritten*
Arabic	1,996	570	4,206	3,370
Bengali	1,608	401	949	8,919
Gujarati	1,229	144	982	37
Gurmukhi	3,629	538	5,475	135
Devanagari	1,706	1,203	1,076	301
Japanese	1,451	352	363	89
Kannada	1,183	872	974	1,123
Malayalam	2,370	575	1,950	144
Oriya	1,660	333	649	7,514
Roman	1,574	558	6,053	3,750
Tamil	451	873	1,667	557
Telugu	1,261	640	865	161
Thai	1,856	1,828	1,861	2,644
Total	21,974	8,887	27,070	28,744
	30,861		**55,814**	
	86,675			

This dataset presents two key challenges and implications. First, the imbalance in the dataset may affect the model's performance across different scripts, with scripts that have fewer training examples potentially resulting in lower accuracy. Second, the unequal data distribution across the training and testing sets may have an impact on the reliability of the evaluation results, making it more difficult to evaluate the model's overall effectiveness accurately.

B. Evaluation metrics

The primary evaluation metric used in the competition was correct classification accuracy (CCA). This metric is determined as the proportion of correctly identified samples divided by the total number of samples for each task. CCA is defined by the following equation:

$$CCA = \left(\frac{N_{correct}}{N_{total}}\right) \times 100 \qquad (1)$$

where Ncorrect represents the number of correctly classified samples, and Ntotal is the total number of samples.

Given the class imbalance in the training and test sets, our proposed method needs to address this challenge effectively to achieve high performance. The competition required robust models that could sustain high accuracy despite imbalanced data, ensuring fair evaluation across various script samples. To further evaluate performance, we use the following metrics:

- *F1 score*: A metric that combines precision and recall to provide a single measure of a model's performance, particularly in imbalanced datasets. The F1 score is calculated using the harmonic mean of precision and recall:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \qquad (2)$$

where Precision is the number of true positive predictions divided by the total number of positive predictions, and Recall (or Sensitivity) is the number of true positive predictions divided by the total number of actual positive instances.

- *Balanced accuracy (BA)*: This metric is particularly helpful when dealing with imbalanced datasets. This accounts for the fact that predicting the majority class can lead to misleadingly high accuracy. The balanced accuracy is calculated by averaging the recall (true positive rate) obtained for each class:

$$BA = \frac{1}{2}\left(\frac{True\ Positive}{True\ Positive + False\ Negative} + \frac{True\ Negative}{True\ \ Negative + False\ Positive}\right) \qquad (3)$$

- *Receiver operating characteristic (ROC) or Area under the curve (AUC) score*: A popular metric for evaluating classification model performance. For multiclass classification, we use the one-vs-one (OvO) method, which calculates the ROC AUC for each pair of classes. If there are k classes, this yields k(k-1)/2 pairs. The final score is the average AUC score across all pairs.

4. Experiments

This section evaluates the effectiveness of our proposed method (YafNet) on three tasks using handwritten and printed word images. We also compare its performance to that of other popular methods for script identification. The three tasks are listed below: Task 1: Script identification in handwritten word images; Task 2: Script identification in printed word images; and Task 3: Mixed script identification, which requires training and testing on both handwritten and printed word images. The experiments in this work have been carried out on an Intel CORE i7 9th Gen processor, 32 GB of RAM, and a GPU with 6 GB of memory.

When developing and training a CNN from scratch for script identification, optimizing hyperparameters is crucial for achieving the best performance. For YafNet, we adopted the Adam optimizer with a learning rate of 0.001 and trained it for 100 epochs with a batch size of 32. To examine the effect of input image size on our model, we evaluated the same model at four different sizes: 32x32, 64x64, 112x112, and 128x128 pixels. Table 3 shows the correct classification accuracy (CCA) results for our method using varying sizes of handwritten and printed word input images.

As the input image size increases from 32x32 to 112x112, performance progressively improves across all tasks. This improvement is especially noticeable in Task 1 (handwritten script identification), where the CCA rises from 75.02% for 32x32 images to 91.29% for 112x112 images. This result is expected since higher resolution images provide more detailed features, which are essential for effective classification. This is particularly true for handwritten data, where subtle details in strokes significantly influence classification accuracy. Interestingly, the CCA drops significantly to 90.87% at 128x128, indicating a potential point of diminishing returns. Larger image sizes may introduce additional noise or extra complexity, which could negatively affect performance.

Table 3: the CCA results for our proposed method using varying sizes

Input image size	Task 1 (handwritten)	Task 2 (printed)	Task 3 (mixed)
32x32x1	75.02%	85.18%	79.95%
64x64x1	86.83%	93.59%	90.11%
96x96x1	90.81%	95.98%	93.32%
112x112x1	**91.29%**	**95.72%**	**93.44%**
128x128x1	90.87%	94.21%	92.49%

The performance in Task 2 (printed script identification) is consistently high, reaching 95.98% at 96x96 and remaining stable at 112x112 and 128x128. This result indicates that printed text benefits from higher resolution but does not yield significant gains beyond a certain threshold. The model obtains near-optimal performance beginning at 96x96, suggesting that printed text is easier to classify once a certain level of detail is captured. Similarly, in Task 3 (mixed script identification), performance improves as image size increases, with a CCA of 93.44% at 112x112. However, performance drops to 92.49% at 128x128, implying that the additional resolution may not give much new information and may even introduce unneeded complexity or noise.

Across all tasks, the 112x112-image size appears to be the optimal input resolution, providing the best balance between capturing sufficient detail and avoiding overcomplexity. This illustrates the need of adjusting the input resolution to the specific characteristics of the data being classified. Table 4 shows detailed performance metrics for the proposed method with 112x112 images across various tasks (handwritten, printed, and mixed).

Table 4: performance metrics for YafNet on a 112x112 basis

Train with	Metric	Test with		
		Handwritten words	*Printed words*	*Mixed words*
	CCA	**90.19%**	35.61%	63.72%
Task 1 *Handwritten words*	F1 Score	**91.12%**	35.93%	66.57%
	BA	**87.51%**	51.69%	65.95%
	ROC AUC Score	**99.15%**	91.06%	95.08%
Task 2 *Printed words*	CCA	36.63%	**91.23%**	63.11%
	F1 Score	41.92%	**92.12%**	66.76%
	BA	34.87%	**96.55%**	74.76%
	ROC AUC Score	81.12%	**99.98%**	97.60%
Task 3 *Mixed words*	CCA	**91.29%**	**95.72%**	**93.44%**
	F1 Score	**91.69%**	**95.90%**	**93.67%**
	BA	**88.63%**	**97.39%**	**95.16%**
	ROC AUC Score	**99.12%**	**99.98%**	**99.83%**

The model achieves a strong accuracy of 90.19% when trained and tested on handwritten words, demonstrating its ability to generalize well to the variability of handwriting. However, when tested on printed words after training on handwritten words, the accuracy drops significantly to 35.61%, indicating that handwritten training data do not translate well to printed word recognition. Similarly, when trained on printed words, the model reaches 91.23% accuracy, but when evaluated on handwritten words, accuracy drops to 36.63%, showing the specificity of training on printed data. Training on both types of data results in balanced performance, with 91.29% accuracy for handwritten words and 95.72% accuracy for printed words, demonstrating the effectiveness of mixed training data in generalizing across all tasks.

For handwritten words, the F1 score is 91.12%, reflecting a good balance between precision and recall. However, the F1 score for printed words in Task 1 is low (35.93%), supporting the fact that training only using handwritten data does not generalize well to printed words. The F1 score for printed words is 92.12%, whereas for handwritten words, it is significantly lower (41.92%), indicating that models trained only on printed data struggle with handwritten data. The F1 score for mixed words shows the best overall performance (93.67%), demonstrating how mixed training data improve both precision and recall across handwritten and printed tasks. When trained on handwritten data, the model achieves 87.51% balanced accuracy but only 51.69% accuracy for printed words. The balanced accuracy is high for printed words (96.55%) but much lower for handwritten words (34.87%) when trained on printed data. The balanced accuracy reaches a peak of 97.39% in Task 3, confirming that training on both handwritten and printed words yields the best overall performance and generalizability. Furthermore, training on mixed data achieves the highest overall ROC AUC scores, 99.12% for handwritten words and 99.98% for printed words.

The metrics show that training on both handwritten and printed data (Task 3) leads to the best generalizability and overall performance across all tasks. Models trained only on handwritten or printed words tend to perform poorly when tested on other data types, illustrating the importance of mixed data training for diverse tasks.

To further improve YafNet's script identification, we implemented both data augmentation (DA) and external data (ED) methods. First, we used data augmentation techniques to increase the variety of the training data while minimizing overfitting. Second, we combined the provided dataset with external datasets to enrich the training set.

- *Data Augmentation*: We applied various transformations to the text images, including rotation, shear, and zoom.
- *External Data*: For the handwritten scripts, we developed a word-level dataset from PHDIndic_11 [21] and BN-HTRd [22]. The dataset includes eight classes: Bangla (6,331 words), Oriya (4,993 words), Roman (2,200 words), Gurmukhi (914 words), Telegu (525 words), Devanagari (481 words), Tamil (361 words), and Kannada (272 words). For the Gurmukhi printed script, we additionally made a dataset of 4,037 words from internet documents. These datasets have been added to the training set.

Table 5 shows YafNet's performance across various tasks—handwritten, printed, and mixed scripts—with data augmentation (DA) and external datasets (ED). YafNet achieves strong baseline performance in all tasks, with 91.29% for handwritten, 95.72% for printed and 93.44% for mixed scripts. This result indicates that YafNet's core architecture is strong in script identification even without augmentation or external data. Applying DA alone resulted in considerably improved performance across all tasks, particularly handwritten scripts (93.77%). Similarly, adopting ED increases CCA to 94.52% for handwritten words, whereas printed and mixed scripts benefit albeit slightly less. The biggest enhancement can be observed when both DA and ED are combined, resulting in remarkable results, with 96.03% for handwritten, 99.52% for printed, and 97.73% for mixed scripts. This result shows that increasing data diversity reduces overfitting and improves generalizability.

Table 5: YafNet performance metrics using DA and ED across all tasks

Proposed method	Metric	Task 1 (Handwritten)	Task 2 (Printed)	Task 3 (Mixed)
YafNet	CCA	91.29%	95.72%	93.44%
	F1 Score	91.69%	95.90%	93.67%
	BA	88.63%	97.39%	95.16%
	ROC AUC Score	99.12%	99.98%	99.83%
YafNet using DA	CCA	93.77%	99.08%	96.35%
	F1 Score	94.38%	99.10%	96.36%
	BA	90.38%	99.48%	96.70%
	ROC AUC Score	99.41%	99.99%	99.92%
YafNet using ED	CCA	94.52%	98.15%	96.36%
	F1 Score	94.83%	98.37%	96.40%
	BA	89.75%	99.11%	96.91%
	ROC AUC Score	99.42%	99.99%	99.94%
YafNet using DA and ED	CCA	**96.03%**	**99.52%**	**97.73%**
	F1 Score	**96.33%**	**99.53%**	**97.73%**
	BA	**92.85%**	**99.69%**	**97.93%**
	ROC AUC Score	**99.65%**	**99.99%**	**99.97%**

The F1 score reflects a strong balance between precision and recall, with 91.69% for handwritten, 95.90% for printed, and 93.67% for mixed scripts. These values demonstrate that YafNet performs well at balancing false positives and false negatives. The combination of the DA and ED raises F1 scores across all tasks. Handwritten script identification benefits the most from DA (94.38%), especially when combined with ED (96.33%). This indicates better handling of imbalanced data and more consistent identification of correct scripts. The F1 score for printed scripts improves from 95.90% to 99.53%, demonstrating that augmenting the data and incorporating external datasets allows the model to balance false positives and false negatives more effectively. The mixed task increases from 93.67% to 97.73%, showing that these techniques help achieve strong overall performance across different script types.

YafNet has a balanced accuracy of 88.63% for handwriting, 97.39% for printing, and 95.16% for mixed scripts. For handwriting tasks, the BA increases to 92.85%. The printed task shows a significant improvement (99.69%), demonstrating the method's strength in distinguishing between printed scripts. In Task 3, the balanced accuracy improved to 97.93%, reflecting strong overall performance on both handwritten and printed scripts. The ROC AUC scores show near-perfect discrimination ability across tasks, particularly when the DA and ED are combined. The mixed task improves from 99.83% to 99.97%, indicating the model's ability to make highly accurate and confident predictions on both handwritten and printed scripts.

The results in Table 5 demonstrate the significant advantages of applying DA and ED to improve YafNet performance. While the model performs well even without these strategies, including them results in significant improvements, particularly for handwritten scripts, where difficulties such as heterogeneity in writing styles are more prominent. The combination of DA and ED improves YafNet's generalization across varied tasks, making it a more versatile model for identifying printed and handwritten scripts.

We proceed to further analyse our proposed method. Figure 4 shows the confusion matrix obtained using YafNet (Task 3). The results demonstrate YafNet's performance in script identification for both handwritten and printed word images across 13 scripts. This matrix effectively visualizes how well the model predicts the correct script (on the diagonal) and highlights the instances of misclassification (off the diagonal).

	Arab	Ban	Guj	Gur	Dev	Jap	Kan	Mal	Ori	Rom	Tam	Tel	Thai
Arab	0.99	0	0	0	0	0	0	0	0	0	0	0	0
Ban	0	0.94	0.01	0.01	0.01	0	0	0	0.02	0	0	0	0
Guj	0	0	0.98	0	0	0	0	0	0	0	0	0	0
Gur	0	0	0	0.99	0.01	0	0	0	0	0	0	0	0
Dev	0	0	0.01	0.01	0.98	0	0	0	0	0	0	0	0
Jap	0	0	0	0	0	0.99	0	0	0	0	0	0	0
Kan	0	0	0	0	0	0	0.95	0	0	0	0	0.02	0.01
Mal	0	0	0	0	0	0	0	1.00	0	0	0	0	0
Ori	0	0	0	0	0	0	0	0	0.98	0	0	0	0
Rom	0	0	0	0	0	0	0	0	0	0.99	0	0	0
Tam	0	0	0	0	0	0	0	0.01	0	0	0.98	0	0
Tel	0	0	0.01	0	0	0	0.01	0	0	0.01	0	0.97	0
Thai	0	0	0	0	0	0	0	0.01	0	0	0	0	0.98

Figure 4. Confusion matrix (Task 3) obtained using YafNet

YafNet performs excellently in distinguishing scripts like Arabic, Gurmukhi, Japanese, Roman and Malayalam, as indicated by perfect scores of 0.99 or 1.00 on the diagonal for these scripts. This demonstrates that the model is highly accurate in identifying these scripts, likely because they possess distinct features that YafNet can capture effectively. Other scripts, including Gujarati (0.98), Devanagari (0.98), Oriya (0.98), Tamil (0.98), Thai (0.98) and Telugu (0.97), exhibit high classification accuracy with moderate confusion. These results indicate that YafNet can identify both handwritten and printed variants of these scripts with remarkable precision. Scripts like Bangla (0.94) and Kannada (0.95) have slight misclassifications. For example, there is some misunderstanding between Bangla and Oriya due to visual similarity between the characters in handwritten forms. Kannada likewise has a lower accuracy, probably because to parallels with other scripts like Telugu and Thai. The low values in the off-diagonal entries indicate that YafNet rarely confuses one script for another, even when dealing with handwritten and printed word images. This outcome validates the model's strength in handling diverse scripts.

To compare our method, we present and analyse the results of existing script identification approaches. In order to offer a fair comparison, we restricted our analysis to methods that used the SIW 2021 competition dataset. Table 6 summarizes the technique abbreviations used in several script identification methods, providing a simple reference for comparing different approaches.

Table 6: summary of the techniques used in the compared methods for script identification. PM = pretrained models, HC = handcrafted features, Pre = pre-processing, Post = post-processing, DA = data augmentation, ED = external data, D&A = detection and alignment, EM = ensemble models, DM = differentiated models.

Method	Used Technique								
	PM	*HC*	*Pre*	*Post*	*DA*	*ED*	*D&A*	*EM*	*DM*
HRnet48, OCRnet and DCN [3]	**Yes**	No	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	No	No
ResNet-101, ResNeSt-200 and DenseNet-121 [3]	**Yes**	No	No	No	**Yes**	**Yes**	No	**Yes**	No
ResNet50 and NFnet-f3[3]	**Yes**	No	**Yes**	No	**Yes**	No	No	No	No
ResNet and EfficientNet [3]	**Yes**	No	No	**Yes**	No	No	No	**Yes**	**Yes**
EfficientNet [3]	**Yes**	No	**Yes**	**Yes**	No	**Yes**	No	No	No
Oriented Basic Image Feature columns [3]	No	**Yes**	No	No	No	No	No	No	No
Dense Multi-Block Linear Binary Pattern [14]	No	**Yes**	No	No	No	No	No	No	No
ResNet Model [14]	**Yes**	No	No	No	No	No	No	No	No
VGG Model [14]	**Yes**	No	No	No	No	No	No	No	No
Our model (YafNet)	No	No	**Yes**	No	**Yes**	**Yes**	No	No	No

Table 7 displays comparison data for the three different tasks. The results are presented in terms of CCA. We can observe that the HRnet48, OCRnet, and DCN combined method consistently achieves the highest scores across all the tasks, owing to its sophisticated structure and the use of multiple enhancement techniques. Their reliance on pretrained models and post-processing gives them an edge in performance. YafNet demonstrates remarkable performance, particularly in printed and mixed tasks, where it ranks third overall. Despite the fact that YafNet does not employ pretrained models (PM), post-processing (Post), ensemble models (EM) or detection and alignment techniques (D&A), its strategic use of DA and ED allows it to achieve competitive results, demonstrating the importance of these strategies in improving model generalization.

Table 7: summary of existing methods' performance for each of the three tasks

Methods	Task 1 (Handwritten)	Task 2 (Printed)	Task 3 (Mixed)
HRnet48, OCRnet and DCN [3]	99.69% (1)	99.99% (1)	99.84% (1)
ResNet-101, ResNeSt-200 and DenseNet-121 [3]	97.80% (3)	99.80% (2)	98.87% (2)
Our model (YafNet)	**96.03% (4)**	**99.52% (3)**	**97.73% (3)**
ResNet50 and NFnet-f3[3]	99.14% (2)	95.06% (8)	97.17% (4)
ResNet and EfficientNet [3]	95.85% (5)	98.63% (5)	97.09% (5)
EfficientNet [3]	90.59% (6)	99.24% (4)	94.79% (6)
Dense Multi-Block Linear Binary Pattern [14]	89.78% (7)	95.51% (7)	94.45% (7)
ResNet Model [14]	87.72% (8)	96.46% (6)	- (8)
VGG Model [14]	77.69% (9)	94.06% (9)	- (9)
Oriented Basic Image Feature columns [3]	81.21% (10)	86.62% (10)	83.83% (10)

The ResNet models perform remarkably well, particularly in the printed task, due to their deep architecture and robust feature extraction capabilities. However, they fall slightly behind YafNet in the mixed task, suggesting that YafNet's combination of DA and ED provides an advantage. Handcrafted feature-based methods have the lowest performance across all tasks. This underlines the superiority of deep learning models in capturing complex patterns and variations in script identification tasks.

The comparison of different approaches demonstrates that while pertained and combined models often lead to high performance, a well-designed CNN like YafNet, which uses data augmentation, can achieve competitive results without the complexity of advanced architectures. YafNet distinguishes itself through its simplicity and efficiency, achieving strong performance without relying on the complexity of pertained models, ensembles, or post-processing methods. This reinforces the value of a data-driven approach, particularly when training from scratch is required or preferable. The results show that although complex models with multiple enhancements perform well, simpler models like YafNet, when optimized correctly, can excel in script identification tasks. Additionally, YafNet's lightweight design reduces training time and hardware requirements, making it more practical for real-world applications. Its strong generalization capabilities and scalability position it as a resource-efficient yet powerful alternative in the field of pattern recognition.

5. Conclusion

In this study, we propose a robust convolutional neural network (CNN) architecture for script identification that was developed entirely from scratch, without relying on pertained models, detection and alignment techniques, ensemble models, or post-processing methods. Our model is methodically crafted using multiple convolutional and pooling layers, as well as batch normalization, to stabilize training and improve performance. The experimental results demonstrate that our approach achieves high classification accuracy, handling both handwritten and printed word images. This achievement underscores our model's effectiveness in addressing script identification issues, including class imbalances and script type variations. The incorporation of data augmentation and external data has proven useful in improving model performance, paving the way for further advancements in document analysis and recognition. Overall, YafNet provides a solid foundation for future research and practical applications, offering a promising solution to address the complexities of script identification in diverse and challenging contexts.

Funding: "This research received no external funding"

Conflicts of Interest: "The authors declare no conflict of interest."

References

[1] Naosekpam, V. & Sahu, N., Text detection, recognition, and script identification in natural scene images: a Review. International Journal Of Multimedia Information Retrieval, 11(3), 291-314, 2022. https://doi.org/10.1007/s13735-022-00243-8

[2] Ubul, K., Tursun, G., Aysa, A., Impedovo, D., Pirlo, G., & Yibulayin, I., Script Identification of Multi-Script Documents: A Survey. IEEE Access, vol. 5, 6546-6559, 2017. https://doi.org/10.1109/access.2017.2689159.

[3] Das, A., Ferrer, M. A., Morales, A., Diaz, M., Pal, U., Impedovo, D., Li, H., Yang, W., Ota, K., Yao, T., Hung, L. Q., Cuong, N. Q., Kim, S., & Gattal, A., ICDAR 2021 Competition on Script Identification in the Wild. In Lecture notes in computer science, 738–753, 2021. https://doi.org/10.1007/978-3-030-86337-1_49

[4] Pati, P. B. & Ramakrishnan, A., Word level multi-script identification. Pattern Recognition Letters, 29(9), 1218–1229, 2008. https://doi.org/10.1016/j.patrec.2008.01.027

[5] Singh, A. K., Mishra, A., Dabral, P., & Jawahar, C. V., A Simple and Effective Solution for Script Identification in the Wild. 2016 12th IAPR Workshop on Document Analysis Systems (DAS), 428-433, Santorini, Greece, 2016. https://doi.org/10.1109/DAS.2016.57.

[6] Boudraa, M., Bennour, A., Al-Sarem, M., Ghabban, F., & Bakhsh, O. A., Contribution to historical manuscript dating: A hybrid approach employing hand-crafted features with vision transformers. Digital Signal Processing, 149, 104477, 2024. https://doi.org/10.1016/j.dsp.2024.104477

[7] Shi, B., Bai, X., & Yao, C., Script identification in the wild via discriminative convolutional neural network. Pattern Recognition, vol. 52, 448–458, 2016. https://doi.org/10.1016/j.patcog.2015.11.005

[8] Bhunia, A. K., Konwer, A., Bhunia, A. K., Bhowmick, A., Roy, P. P., & Pal, U., Script identification in natural scene image and video frames using an attention based Convolutional-LSTM network. Pattern Recognition, vol. 85, 172–184, 2019. https://doi.org/10.1016/j.patcog.2018.07.034

[9] Khalil, A., Jarrah, M., Al-Ayyoub, M., & Jararweh, Y., Text detection and script identification in natural scene images using deep learning. Computers & Electrical Engineering, vol. 91, 107043, 2021. https://doi.org/10.1016/j.compeleceng.2021.107043

[10] Zhang, Z., Eli, E., Mamat, H., Aysa, A., & Ubul, K., EA-CONVNEXT: An approach to script identification in natural scenes based on edge flow and coordinate attention. Electronics, 12(13), 2837, 2023. https://doi.org/10.3390/electronics12132837

[11] Li, X., Zhan, H., Shivakumara, P., Pal, U., & Lu, Y., SANet-SI: A new Self-Attention-Network for Script Identification in scene images. Pattern Recognition Letters, vol. 171, 45–52, 2023. https://doi.org/10.1016/j.patrec.2023.04.015

[12] Peng, F., Ma, H., Liu, L., Lu, Y., & Suen, C. Y., Adaptive feature fusion for scene text script identification. Multimedia Tools and Applications, 83(23), 62677–62699, 2024. https://doi.org/10.1007/s11042-023-17986-z

[13] Gupta, M. K., Dhawan, S., & Kumar, A., Document Image Script Identification using Deep Network. 2024 11th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 174-179, 2024. https://doi.org/10.1109/SPIN60856.2024.10511557

[14] Ferrer, M. A., Das, A., Diaz, M., Morales, A., Carmona-Duarte, C., & Pal, U., MDIW-13: a New Multi-Lingual and Multi-Script Database and Benchmark for Script Identification. Cognitive Computation, 16(1), 131–157, 2024. https://doi.org/10.1007/s12559-023-10193-w

[15] Jindal, A., Script identification in handwritten and printed documents using convolutional recurrent connection. Multimedia Tools and Applications, 2024. https://doi.org/10.1007/s11042-024-19106-x

[16] Boudraa, M., Bennour, A., Mekhaznia, T., Alqarafi, A., Marie, R. R., Al-Sarem, M., & Dogra, A., Revolutionizing Historical Manuscript Analysis: A Deep Learning Approach with Intelligent Feature Extraction for Script Classification. Acta Informatica Pragensia, 13(2), 251–272, 2024. https://doi.org/10.18267/j.aip.239

[17] Abbas, F., Gattal, A., & Menassel, R. Local binary pattern and its derivatives to handwriting-based gender classification. Bulletin of Electrical Engineering and Informatics, 12(6), 3571-3583, 2023. https://doi.org/10.11591/eei.v12i6.5488

[18] Gattal, A. & Abbas, F., Isolated handwritten digit recognition using LPQ and LBP features. In Proceedings of the 10th International Conference on Information Systems and Technologies, 1-5, 2020. https://doi.org/10.1145/3447568.3448465.

[19] Abbas, F., Gattal, A., Djeddi, C., Siddiqi, I., Bensefia, A., & Saoudi, K., Texture feature column scheme for single- and multi-script writer identification. IET Biometrics, 10(2), 179–193, 2021. https://doi.org/10.1049/bme2.12010

[20] Zhang, Z., Mamat, H., Xu, X., Aysa, A., & Ubul, K., FAS-Res2Net: an improved ReS2Net-Based script identification method for natural scenes. Applied Sciences, 13(7), 4434, 2023. https://doi.org/10.3390/app13074434

[21] Obaidullah, S. M., Halder, C., Santosh, K. C., Das, N., & Roy, K., PHDIndic_11: page-level handwritten document image dataset of 11 official Indic scripts for script identification. Multimedia Tools and Applications, 77(2), 1643–1678, 2018. https://doi.org/10.1007/s11042-017-4373-y

[22] Rahman, M. A., Tabassum, N., Paul, M., Pal, R., & Islam, M. K., BN-HTRD: a benchmark dataset for document level offline Bangla handwritten Text recognition (HTR) and line segmentation. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2206.08977