ISSN: 0067-2904

Advancing Arabic Text Detection: A CNN-Heuristic and EfficientNet-B1-Based Framework

Aissa Amrouche
Department of Electronics, Faculty of Technology, University of Blida 1, Blida, Algeria.
Computational Linguistics Department, Scientific and Technical Research Centre for the Development of Arabic Language CRSTDLA, Algeria.

Received: 30/3/2025 Accepted: 12/10/2025 Published: 30/ 8/2026

Abstract

In computer vision, word detection in images is still a major challenge, especially for applications like scene interpretation, document indexing, and machine translation. Arabic script presents particular difficulties because of its cursive style, the use of diacritical marks, and the range of typefaces and orientations, which sometimes lead to errors in standard models, even though this effort has achieved great strides in Latin scripts. In this work, we propose a two-step method for Arabic Text Detection (ATD), aiming to improve accuracy without overly complex pipelines. Our approach starts with a set of heuristic rules to pre-select candidate regions. These rules rely on basic geometric and statistical cues, which helped us eliminate much of the irrelevant background noise in initial tests. To further improve the choices, we use an EfficientNet-B1 CNN. This model uses Global Average Pooling and a final Softmax layer to distinguish between text and non-text regions. The evaluation, which was conducted on a broad dataset of Arabic scene photos, revealed that combining heuristics with CNN verification enhances results significantly. For instance, we observed an increase in detection accuracy from 89.45% (heuristics alone) to 96.46% with the full pipeline. Moreover, the CNN classifier itself reached an accuracy of 97.77%. These findings confirm the robustness of our approach, particularly in cluttered visual environments.

Keywords: Text detection, Object detection, Text localization, CNN, Heuristic rules, OCR, Arabic text, EfficientNetB1.

1. Introduction

Text detection (TD) is an important component in state-of-the-art recognition systems in various applications, including web searching, document categorization, and visual content retrieval. It has seen increasing relevance in recent years in various applications such as autonomous driving, assistive technology for blind users, and even augmented reality systems. Despite this, automatic text spotting in natural scene images remains a challenge [1].

There are a number of challenges: font changes, non-uniform layouts, changes in illumination, and backgrounds with heavy clutter. These problems can obscure or warp the text regions, particularly in cluttered environments such as city streets or commercial places.

*Email: aissa_amrouche@univ-blida.dz

When TD does not segment images properly, the results of subsequent stages, especially OCR, are significantly affected. The perception of natural environments introduces additional uncertainty. The presence of disruptive elements and the lack of structural consistency in advertisements and storefront signage often necessitate more resilient methods than conventional geometric techniques [2].

There are many factors that can reduce the clarity of a video, including motion blur, low spatial resolution, and compression, which may likely add to the task complexity. The Arabic script offers its own set of challenges. The cursive script and diacritical marks with many ligatures increase the probability of misrecognition, especially when character fragments are crossed or distorted. The complexity is compounded by changes in illumination, occlusion, and distorted viewpoints [3].

This study seeks to create an automated system capable of detecting and localizing Arabic text within images across various conditions, such as inadequate lighting, low contrast, and reduced resolution. To accomplish this, we introduce a hybrid methodology that combines traditional Heuristic Rules (HR), including dilation, smoothing, and projection techniques, with contemporary deep learning methods. Our methodology specifically employs Convolutional Neural Networks (CNNs) based on the EfficientNet-B1 architecture. While deep learning has shown considerable success in fields such as speech and image processing, its use in Arabic text detection remains largely unexplored. CNNs offer the advantage of acquiring hierarchical features from raw input data, which allows the model to more efficiently distinguish between textual and non-textual regions. These features are especially advantageous when addressing the intricate structures of Arabic script.

The organization of the paper is as follows. Section 2 details the primary motivations and contributions of our research. Section 3 examines related literature in the domain of text detection. In Section 4, we elaborate on the proposed detection methodology. Section 5 outlines the experimental framework and presents the evaluation findings. Lastly, Section 6 wraps up the paper and proposes avenues for future investigation.

2. Motivation and Contributions

2.1 Motivation for Arabic Text Detection

With 22 countries and more than 420 million speakers, the Arabic language is one of the major languages, and thus Arabic Text Detection (ATD) becomes an interesting area, which has various potential applications such as document analysis, machine translation, and assistive systems [4]. However, Arabic text recognition in images presents its own challenges. The script is cursive, with letter shapes that can change depending on their position in a word. It is a right-to-left script, and diacritics may be used to pad characters [5].

Additionally, background noise, poor image quality, and different writing forms make detection more difficult and may degrade accuracy and efficiency. Although numerous studies have strived to enhance ATD (mainly aiming at trying to solve these challenging problems), existing approaches have their limitations in addressing these challenges, such as high FP rates and incomparability across datasets. In this paper, we develop a robust detection framework that employs heuristic pre-processing and CNN-based feature extraction. This combination of methods increases both the accuracy and robustness of the system. Thus, it is better for processing diverse and severe practical images.

2.2 Contributions

This paper introduces two complementary approaches in order to enhance the detection of Arabic text in images:

1- CNN-Heuristic Rules (CNN-HR) Hybrid: This method is a hybrid technique using conventional heuristic rules to search for possible text and then applying CNN to verify the found regions. The CNN assists in sharpening the output by eliminating the false positives. This paper presents two complementary methods designed to improve the detection of Arabic text in images.

2- Deep learning EfficientNet B1 model: With this model, we were able to lean heavily on the robust feature extraction and excellent classification capability of the efficientNetB1 architecture, leading to more reliable detection under loose conditions.

The main contributions of this study are:

- Advancement Detection Accuracy: You can easily read your tire condition after using this new tire pressure gauge.
- The CNN-HR method achieves overall accuracy improving from 88.31% to 96.46%.
- The classification accuracy obtained by the EfficientNetB1 model is 97.77%, which is superior to the state-of-the-art available methods.
- Resilience in the presence of complexity:
- The system performs well on real-world images with various fonts and low-resolution and cluttered backgrounds.
- Comprehensive Needs Assessment:
- Model performance was validated using Performance metrics such as Precision, Recall, F1-score, and Overall Accuracy and the result established the superior detection of the models.
- Comparison with Prior Work:
- The presented models achieve higher precision (98.72%) and recall (98.93%) compared to existing models available in literature.
- Future Research Recommendations:
- This work paves the way to adopting transformer-based architectures in Arabic text detection and moving closer to the development of end-to-end Arabic character recognition systems.

By combining heuristic-based filtering and state-of-the-art deep-learning networks, the proposed method provides a more accurate, reliable, and time-efficient Arabic text-detection solution with respect to the previous methods.

3. Related Works

Text detection (TD) in images is one of the core topics in computer vision, with many existing algorithms developed for extracting textual information in visual data [6]. Text in images plays an important role in many applications as it conveys semantic information that is important for scene understanding and context modeling. Instead of purely attending to the text positions, some methods use semantic cues to localize potential text regions. When such regions are separated out, character classification algorithms are employed to improve detection and localization accuracy [3]. Numerous studies have been conducted to analyze TD in both images and video, including edges, gradients, colors, textures, and spatial patterns. Such visual cues are the building block of several detection mechanisms, such as machine learning, heuristic rules (HR), and hybrid approaches. Heuristic methods use hard rules (mostly geometric or statistical based) to segment and recognize text. However, ML models reliably learn to discern text from non-text regions after training with labeled datasets. Hybrid methods have attempted to combine the strengths of heuristic approaches and ML models to improve robustness and accuracy. Here, we classify the related work into

three main classes: (1) heuristic-driven solutions, (2) solutions based on machine learning and (3) amalgamated methods involving both paradigms.

3.1 Text detection using heuristic approaches

To locate text in images, heuristic algorithms employ predetermined low-level visual criteria. To segment probable text portions, these algorithms commonly use edge distribution, intensity, density, and structural link. As a result, they are sometimes referred to as Connected Component (CC) or Linked Component (LC) approaches [7]. LC approaches combine pixels into cohesive text-like fragments based on visual characteristics such as color uniformity, edge consistency, and spatial arrangement. One frequent application of Canny Edge Detection (CED) is to locate the edges of potential text. Morphological techniques such as dilatation and opening are then applied to refine these edges. These processes help to improve connectivity and reduce noise, resulting in more clearly delineated text parts [8]. By projecting these features along horizontal and vertical axes, we can improve bounding box accuracy by reducing false detections and reassembling broken characters. The full CED pipeline involves the following steps:

- Gaussian filtering to smooth the image and suppress noise artifacts [9].
- Sobel operator to compute edge gradients in multiple directions.
- Non-maximum suppression to retain only the strongest and most relevant edge pixels.
- Double thresholding to differentiate between strong and weak edges, followed by edge tracking by hysteresis to complete the edge map [10].

Some techniques, such as the one proposed in [11], use the Laplacian operator to identify changes in brightness across the image and then k-means clustering to refine the probable regions. Other heuristic methods that improve text segmentation include Local Binary Patterns (LBP), Discrete Cosine Transforms (DCT), Wavelet Transforms, and Sobel filters [12]. Certain tweaks to heuristic techniques have shown promise in Arabic and Farsi scripts. In [13], a corner-density-based technique used the structure of Arabic script to improve text localization in video frames by focusing on stroke distribution. Despite being quick and cheap to compute, heuristic approaches are often not particularly trustworthy in real life, especially when there are elaborate backdrops, inconsistent lighting, or a wide range of font styles and sizes.

3.2 ML- methods for text detection

Convolutional Neural Networks (CNNs) have become the state-of-the-art technology in text detection (TD) tasks in recent years, far exceeding previous solutions. Pixel-based TD methods and Regression-based TD methods are spread in two main branches. Pixel-level algorithms treat the problem of detection as a classification at the pixel or small region level. Text can potentially be separated from the background with high accuracy using Fully Convolutional Networks (FCNs) that are commonly used for dense, pixel-wise prediction. By explicitly predicting bounding boxes, CNNs are trained to locate text in regression-based techniques, which frame text detection as an object detection (OD) task. Several object detection models have been effectively adapted for TD, including Region-based CNNs (R-CNN, Fast R-CNN, and Faster R-CNN) [13, 14]. These models often use Region Proposal Networks (RPNs) to produce candidate text regions, which are then identified and refined by CNN-based modules. Other important methods are based on Sliding Window (SW), which scans the image at multiple scales and uses a CNN to detect text of different scales and resolution. Classic machine learning methods have also been investigated, in addition to the deep learning methodologies. For instance, in video frames, a pixel-wise classification technique has been performed using Support Vector Machines (SVMs) [15], while Multi-Layer Perceptrons (MLPs) have been used in different tasks to separate text and non-text

[16]. While these machine-learning algorithms have demonstrated excellent performance, they often require large, labeled datasets and strong computing resources, raising concerns when it comes to real-time or resource-limited scenarios.

3.3 Hybrid TD methods

Hybrid TD techniques seek to take advantage of the complementarity of Heuristic Rules (HR) and Machine Learning (ML) approaches. Heuristic-based methods usually obtain a first degree of text candidates, and the filtering is performed by ML classifiers to remove false positives and to refine the localization. For example, the authors in [17] presented a hybrid procedure combining energy-based wavelet transform with density-based region growth to segment text in video materials. In another related study, [14] used Maximally Stable Extremal Regions (MSERs) with Convolutional Neural Networks (ConvNets) for text spotting in challenging natural scenes. Hybrid techniques are also promising in Arabic text detection. In [7], three machine learning (ML) models were proposed to identify Arabic script in news broadcasts and in [15] , a GA-CNN hybrid model for robust text classification was shown.

More recently, [16] proposed a RetinaNet-based Arabic TD system for identifying text in video streams. Motivated by these improvements, in this paper, we propose a new hybrid CNN-Heuristic framework model, which is a combination architecture of EfficientNet-B1 and the classical preprocessing strategies, such as the Sobel filter, and morphological operations. It enables the system to distinguish efficiently between text and background noise through a combination of structural enhancement and deep feature learning. Despite the tremendous improvements that the field has witnessed, many algorithms still face difficulties dealing with Arabic script due to known factors, especially in cluttered backgrounds or images that are acquired under low quality. Although heuristic approaches are usually interpretable and computationally fast, they often fail to be robust across a variety of scenarios. Deep learning methods, on the other hand, reach high accuracy but require large, labeled datasets and computational power.

To address this, we proposed a hybrid architecture using heuristic preprocessing combined with CNN for detection under a variety of realistic scenarios. The methodology that is proposed is described in the next section, which involves the preprocessing steps, feature extraction, and network architecture.

4. Methodology

Figure 1 shows the structure of the proposed method for Arabic text detection and localization in images. The system is composed of two cooperating subsystems for enhancing detection accuracy and robustness. Algorithm 1 presents an overview of the primary steps in our methodology. It illustrates that we first preprocess the data through sequential heuristics and subsequently verify it in parallel through CNN and EfficientNetB1 classifiers.

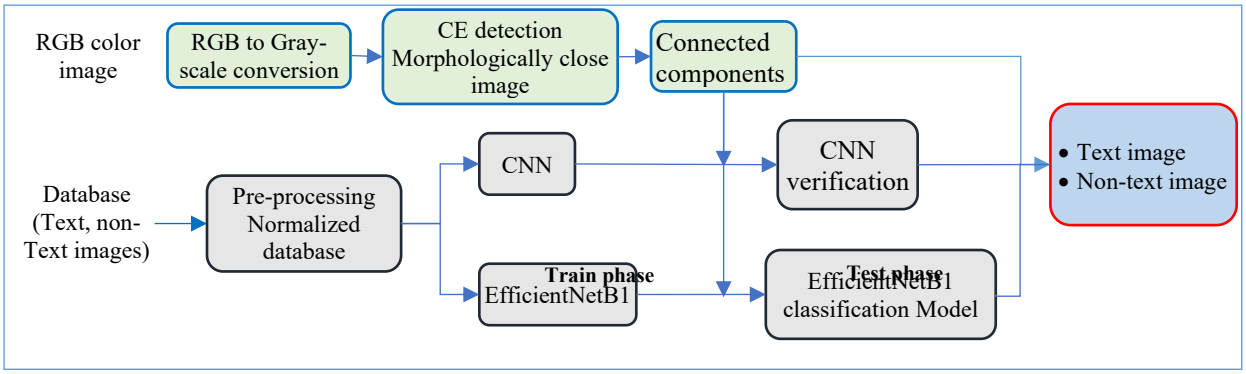

Figure 1: Flowchart of a text detection system.

We describe a two-subsystem solution, where the first subsystem adopts a two-stage method for detecting candidate sections of text based on heuristic pre-processing. These candidate regions are then inputted into a CNN module, which serves as a verifier to determine whether the region is text or non-text. This stage of validation reduces false positives a lot while also making localization more accurate. EfficientNet-B1, a deep neural network made to assist with efficient and effective feature extraction, is part of the second subsystem.

Algorithm 1: Multi-Stage Arabic Text Detection with Heuristic Filtering and Parallel Deep Learning Verification

Input: RGB image
Output: Text / Non-text classification from three independent stages
1. Convert the input RGB image to grayscale
2. If image size > 950 × 512 pixels, proportionally resize
3. Apply Gaussian filtering to reduce noise
4. Apply Canny Edge Detection (CED)
5. Apply morphological closing to enhance edge connectivity
6. Perform Connected Components (CC) segmentation
7. For each connected component:
a. Extract geometric and structural features
b. Apply heuristic filtering using aspect ratio, stroke consistency, and morphological clean-up
8. Store remaining regions as candidate text areas
9. Pass each candidate region in parallel to:
a. A CNN-based verifier
b. An EfficientNetB1 classifier
10. Independently output three classification results:
a. From heuristic filtering
b. From CNN model
c. From EfficientNetB1

Before being used, the model goes through task-specific fine-tuning with a special training module that lets it adapt to various visual situations, like low resolution, complicated backgrounds, and changing lighting. This two-path method makes sure that it functions better in a wider range of visual situations and is more accurate.

4.1 Arabic TD using CNN-Heuristic Rules

For improved Arabic text recognition, RGB images are first converted to grayscale to enable more effective segmentation. If the image size is larger than a predetermined threshold value (950×512 pixels), the image is proportionally reduced to prevent large-scale

computations and ensure uniform input size. Text candidate regions are partitioned through Connected Components (CC) segmentation. The resulting connected components are then filtered using bounding box properties to discard non-text elements. Remaining CCs are grouped into probable text sequences, which are further segmented into word-level patches. Additional geometric analysis of the CCs, such as size, spacing, and alignment, enables the exclusion of unlikely text patterns, improving detection precision.

To continue discriminating among the candidates, a set of heuristic rules is applied:

- Aspect ratio filtering removes components with unrealistic width-to-height ratios.
- Stroke width variation analysis rejects regions with a non-uniform stroke width.
- Edge density analysis ensures adequate structural complexity typical of text content.

Next, morphological operations, i.e., dilation and erosion, are applied for the improvement of character continuity and removal of false positives by removing irregularities and filling gaps in broken strokes. The filtered results are subsequently passed through a CNN-based model validation step that reinforces the presence of text and removes any lingering non-text areas, making the detection pipeline robust and accurate.

4.2 Validation through CNN

To improve accurate categorization and reduce false positives, we use a Convolutional Neural Network (CNN) to look for semantically sound text passages.

The architecture consists of four sequentially active convolutional layers. Each layer is intended to progressively pay attention to increasingly abstract features and permit text to be distinguished from non-text areas (see Figure 2). The purpose of such layers is to emphasize important patterns of Arabic scripts, including object spatial layout, edge connectivity, and stroke arrangement. Thus, the detection method is more precise in general.

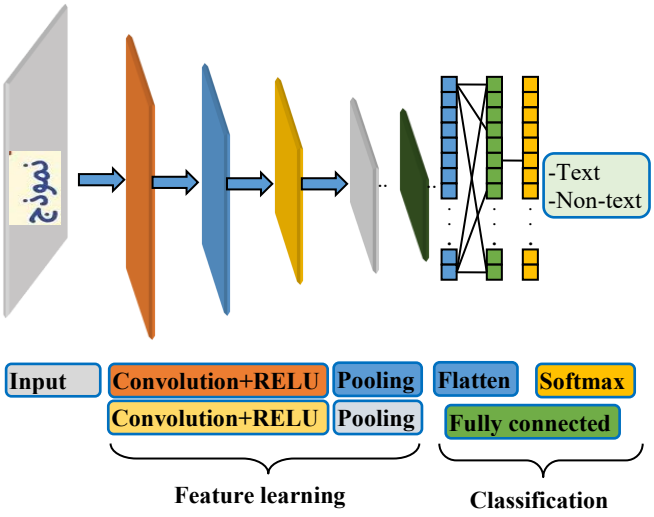

Figure 2: Text verification using the CNN model architecture.

The verification is comprised of some relevant steps that can successfully differentiate between text and non-text regions:

- Convolutional Layers: The layers use small 3×3 filters to build up fine-detailed feature maps, which map the characteristic pattern of text content effectively.
- ReLU Activation: The Rectified Linear Unit (ReLU) introduces non-linearity into the network, enabling it to model complex patterns beyond linear relationships.

- Max-Pooling Layers: These layers decrease the spatial size of feature maps while preserving the most informative characteristics, in order to reduce noise and the number of operations.
- Feature learning layers: The network flattens feature maps and then treats them with fully connected layers to categorize them in terms of the high-level reasoning.
- Softmax Classifier: The last output layer is a vector, and we then apply softmax to get a probability distribution over the classes

In particular, the first convolutional layer (Conv1) takes as input images of size 224×224×3 and uses filters of size 3×3. The feature maps from output are subsequently improved using additional convolutional and pooling layers that allow the network to learn more profound Arabic text hierarchies. Making the network deeper systematically improves the ability to extract hierarchical features, and thus validation accuracy. Table 1 shows the detailed architecture of the proposed CNN-HR module used to check candidate text regions. This helps to make the structure of the module clearer. This CNN model was made to tell the difference between Arabic text and other things by learning about spatial and structural features through a series of convolutional and pooling layers. Each layer helps to improve the feature representation bit by bit, which makes it possible to validate detected regions more accurately. The CNN is trained from a specified dataset for the detection of Arabic text, adapting the model parameters to become as robust as possible and as low as possible in false text detection of images of various categories.

Table 1: CNN-HR Model Architecture and Parameters

Layer	Type	Filter Size	Output Shape	Activation	Remarks
Input	—	224×224×3	224×224×3	—	Input image patch
Conv1	Convolutional	3×3×32	224×224×32	ReLU	Extract low-level features
MaxPool1	Max Pooling	2×2	112×112×32	—	Downsampling
Conv2	Convolutional	3×3× 64	112×112×64	ReLU	Capture mid-level features
MaxPool2	Max Pooling	2×2	56×56×64	—	Further downsampling
Conv3	Convolutional	3×3×128	56×56×128	ReLU	Extract more complex structures
MaxPool3	Max Pooling	2×2	28×28×128	—	Reduce dimensionality
Conv4	Convolutional	3×3×128	28×28×128	ReLU	Deeper feature abstraction
Flatten	—	—	100352 (1D)	—	Prepare for classification
FC1	Fully Connected	128 units	128	ReLU	Learn high-level semantics
FC2	Fully Connected	2 units	2	Softmax	Output: text vs non-text

4.3 TD using EfficientNetB1

EfficientNetB1, a convolutional neural network architecture known for scaling depth, width, and resolution evenly, is applied to text detection [18]. EfficientNetB1 systematically optimizes the dimensions to provide more precision with fewer parameters and lower computational cost than regular CNNs. As can be seen in Figure 3, EfficientNetB1 adopts the typical CNN pipeline of convolutional layers, pooling, batch normalization, activation functions, and fully connected layers. This configuration makes it possible for EfficientNetB1 to derive hierarchical features essential for text area detection, thereby improving detection robustness and accuracy.

Transfer learning is extremely important for vision recognition tasks, especially when labeled data are scarce. We use EfficientNetB1, a strong pre-trained deep model, as an efficient feature extractor in our method. Its pre-training on large image datasets (e.g., ImageNet) provides strong generalized representations that accelerate convergence and

enhance text discrimination capability, especially for Arabic script. Through transfer learning, our system can recognize things with high precision using less data and computational resources compared to training a network from the beginning.

Our results demonstrate that fine-tuning pre-trained models like EfficientNetB1 work very well when it comes to the detection of characters and scene text. This is supported by recent studies [19] showing it works well under similar settings.

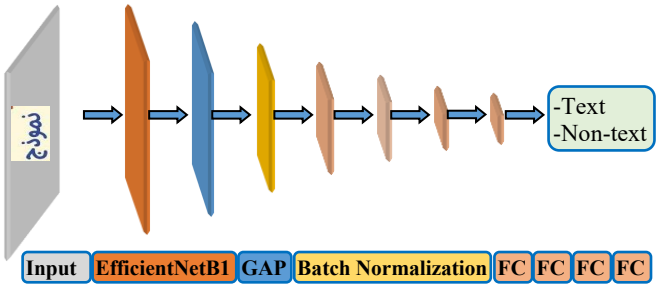

Figure 3: The proposed EfficientNetB1 model.

4.3 1 Architecture and advantages.

EfficientNetB1 uses compound scaling, a novel technique that decreases the network depth, width, and the input image resolution simultaneously in an effort to match computation costs with the efficiency of the model. Equilibrium scaling has many advantages:

- Enhanced parameter effectiveness: EfficientNetB1 achieves equal or even higher accuracy using fewer parameters than typical deep CNN models, thus decreasing computational complexity.
- Better generalization: Through successful extraction of discriminative features, the model copes better with diverse text styles and complex image backgrounds, which is extremely crucial in the case of robust Arabic text detection.
- Lightweight structure: Its light design architecture makes EfficientNetB1 especially suitable for application on resource-limited devices such as mobile devices and embedded systems.

4.3 1 Training Process.

For the purpose of gaining strength against many features in text, EfficientNetB1 is over-trained with a range of techniques:

- Pre-training using pre-trained ImageNet weights that enhance convergence and make the model robust at differentiating features.
- Training instance diversification methods, such as random cropping, contrast, and rotation that increase training instance diversity and generalization to different conditions.
- Adaptive optimization algorithms such as Adam that subtly adjust network parameters to improve accuracy.

The training phase involves incrementing epochs gradually to guarantee successful convergence and prevent under-fitting to allow the model to correctly recognize Arabic text across different font styles, sizes, and complexities in background noise. Utilizing EfficientNetB1's structure and training timetable, the proposed system in this paper achieves optimal computation cost-prediction accuracy balance and is, therefore, efficient and deployable for Arabic text detection from real-world images.

4.4 Database

950 natural scene images were collected, comprising 850 text images and 100 non-text images. The diversity of the quality, contrast, size, orientation, and background complexity of these images exists. These images were laboriously annotated and processed to generate high-quality ground truth for training and evaluation. Images were captured from numerous environments, such as streets in urban areas, shopfronts, signboards, and handwritten documents, over a wide range of real-world appearance conditions.

To facilitate effective training and validation of the model, the data was divided as follows:

- 70% for training: This batch was utilized to train both the CNN-based and EfficientNetB1 models so that they could learn robust text features.
- 30% for testing: This was reserved for testing whether the models could generalize, i.e., recognize Arabic text in unseen images.

In addition, the dataset was pre-processed with image resizing to standard dimensions, pixel value normalization, and text vs. non-text sample balancing. The system outputs two classes (text and non-text) with precise Arabic text localization and recognition.

4.5 Environment and parameters

The system was constructed using Python 3.8, using OpenCV for image preprocessing and segmentation, and TensorFlow for the CNN-based validation module. Experiments were conducted on a workstation equipped with an NVIDIA Quadro M1200 GPU, an Intel Core i7-7700 CPU (2.80 GHz), and 16 GB of RAM.

The essential training parameters were:

- Initial learning rate (LR): 0.001
- LR decay factor: 0.2
- LR decay epochs: 5
- Total epochs: 50
- Mini-batch size: 64

The hyperparameters were trained empirically to provide the best possible balance between the model convergence rate and its detection capability.

4.5.1 Training Results of CNN-HR

Figure 4 shows the CNN-HR model's training and validation loss and accuracy over 50 epochs. At first, both loss curves drop sharply, which means that the model is getting better at telling the difference between text and non-text areas. Early on in training, the training loss drops to about 0.32, and the validation loss does the same thing, which means that the model is likely to converge steadily. The accuracy curves, on the other hand, keep getting better. Training accuracy steadily rises and gets close to 98%, while validation accuracy stays high at about 95%. The training and validation metrics are very similar, which shows that the model is good at generalizing and has a low risk of over-fitting. Overall, the model performs well on previously unseen data.

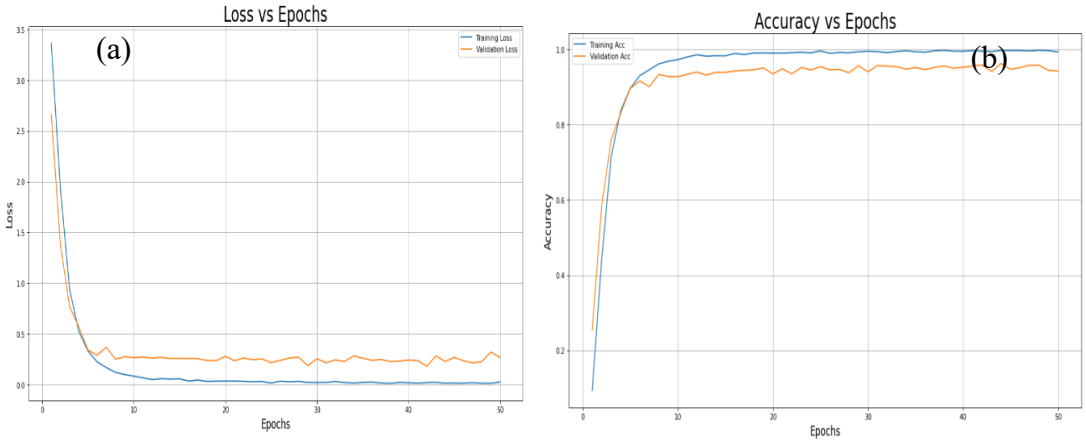

Figure 4: Model Performance:(a) Loss over epochs, (b) Accuracy and loss trends.

4.5.2 Training Results of EfficientNetB1

Figure 5 presents the validation loss of the EfficientNetB1 model introduced in the process of training. The graph shows a dramatic drop in mean loss during the first 10 iterations, declining to around 0.25. After this early drop, the rate of decrease slows and levels off at around 0.08 toward the end of training. This indicates that the model has reached convergence, and further loss improvement from this point forward would be negligibly small. Cumulatively, the plot conveys how effective the training process is in terms of optimizing the performance of the model.

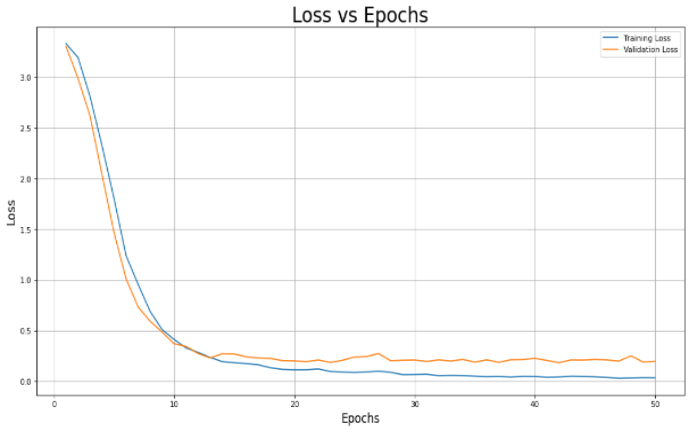

Figure 5: Validation loss during the learning process.

4.5.3 Output Fusion and Voting Strategy

The system uses a parallel validation method to make it robust. According to this method, EfficientNetB1 and CNN-HR modules independently verify candidate text regions. The system does not depend solely on the output of one model; instead, it cross-verifies the findings with the other one and keeps only those detections that are found by both models or with strong evidence in support of one model. This process of joint decision-making works as a soft voting system, which improves detection and there are fewer false positives. The classification results are combined afterwards after these separate steps.

4.6 Evaluation metrics

The experiments were conducted under varying conditions to evaluate the overall performance of the system in terms of the usage of images derived from academic literature

and natural images. The evaluation was conducted on four parameters: precision (Pr), recall (Re), F1-score (F1), and overall accuracy (OA) [20]. The parameters are:

$$\text{precision} = \frac{TP}{TP+FP} \quad (1)$$

$$\text{recall} = \frac{TP}{TP+FN} \quad (2)$$

$$\text{f1} - \text{score} = 2 * \frac{\text{recall} * \text{precision}}{\text{recall} + \text{precision}} \quad (3)$$

$$\text{OA} = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

Where: TP (True Positives): Correctly detected text regions, TN (True Negatives): Correctly classified non-text regions, FP (False Positives): Non-text regions incorrectly detected as text, and FN (False Negatives): Text regions that were not detected.
Precision (Pr) calculates the number of text regions correctly identified out of all regions that were recognized by the system. Recall (Re) is the ratio of true text examples that are being accurately labeled by the system. F1-score is the harmonic mean of precision and recall and gives an estimate of the model's accuracy in an equilibrium manner. OA stands for overall accuracy and is used for the ratio of accurately classified text and non-text areas out of total detections.
Figure 6 depicts the model's performance, demonstrating: Training-related accuracy and loss patterns, Precision performance throughout epochs, and Recall performance throughout epochs.

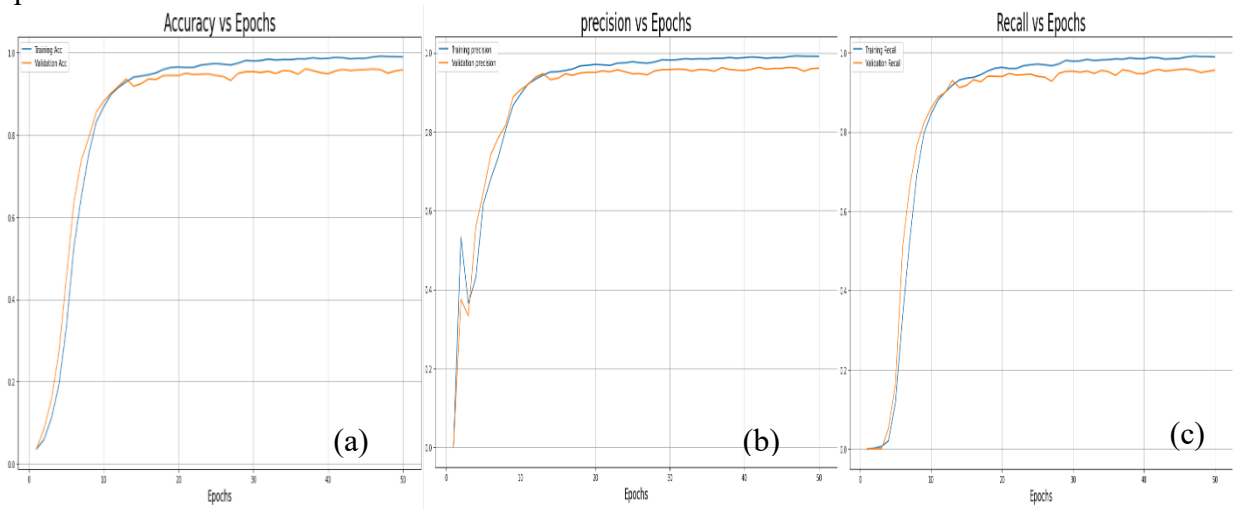

Figure 6: EfficientNetB1 model performance:(a) Accuracy/loss trends, (b) Precision/epochs, (c) Recall/epochs.

5. Simulation results and discussions

In order to evaluate its performance, the proposed text detection system was tested on actual images along with document samples. This section provides the results obtained, quantitatively compares, and benchmarks the performance of the system against other methods.

5.1 Real-World Image Detection Results

Figure 7 illustrates the text detection results for an image sourced from the Al Jazeera television channel. The subfigures show:
(a) Original Image – The raw input containing Arabic text.
(b) Heuristic Rule-Based Detection – Initial text region identification using heuristic rules.

(c) Detection Using EfficientNetB1 – Final detection results produced by the EfficientNetB1 model, demonstrating enhanced text localization accuracy.

Figure 7: Real-world text image detection.

The detection result for this image is the following:

- With only heuristic rules, Overall Accuracy (OA) is 89.47%, whereas Precision (Pr), Recall (Re), and F1-score are both 94.12%.
- The incorporation of the CNN-HR verification system into OA is hugely beneficial, with the OA rising to 97.12%, with corresponding augmentation in Precision, Recall, and F1-score.
- The EfficientNetB1 model achieves the optimal values of 100% for OA, Precision, Recall, and F1-score, thus effectively demonstrating that deep learning is superior to heuristic approaches.

These findings show the extent to which CNN-based verification and EfficientNetB1 significantly improve Arabic text detection accuracy.

5.2 Document Image Detection Results

Figure 8 shows the detection results of two document images. Each word is detected correctly, which verifies the robustness of the proposed approach in the case of structured document images.

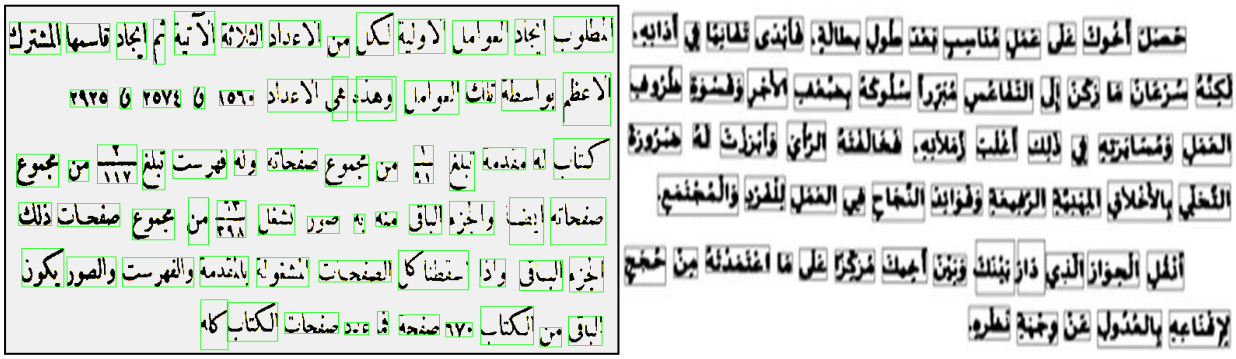

Figure 8: Document text image detection results

Table 2 provides a complete examination of Pr, Re, F1-score, and OA across six different real-world images. The results show that although heuristic rules are a robust baseline, employing CNN-based verification and EfficientNetB1 significantly enhance detection accuracy through efficiently removing false positives and false negatives.

Table 2: Detection Efficiency

Images		Image1	Image2	Image3	Image4	Image5	Image6
Heuristic rules	Pr	88.23	96.15	93.75	95.23	97.56	94.91
	Re	96.77	98.03	93.75	97.08	97.56	94.91
	f1	92.30	97.08	93.75	96.15	97.56	94.91
	OA	86.48	94.54	89.47	93.10	95.74	91.04
CNN verification	Pr	100	98.14	94.73	99.06	100	98.33
	Re	98.64	100	100	99.06	100	100
	f1	99.31	99.06	97.29	99.06	100	99.15
	OA	98.66	98.18	94.73	98.30	100	98.46
EfficientNetB1	Pr	100	99.00	96.77	99.13	100	98.42
	Re	99.00	100	100	99.13	100	100
	f1	99.50	99.50	98.60	99.13	100	99.20
	OA	99.01	99.03	96.84	98.30	100	98.48

5.3 Overall System Performance Analysis

The overall performance of the system over various test scenarios is as follows:

- More than 10 test cases have been executed by the CNN-HR model, where it identified 920 true positives (TP) and 35 true negatives (TN), and produced 25 false positives (FP) and 35 false negatives (FN).
- This resulted in an Overall Accuracy (OA) of 96.46%, with precision (Pr) at 97.35%, recall (Re) at 98.92%, and an F1-score of 98.13%.
- With EfficientNetB1, true positives increased to 931, though true negatives slightly dropped to 27, leading to a reduction in false positives (14) and false negatives (8).
- EfficientNetB1 achieved an OA of 97.77%, precision of 98.72%, recall of 98.93%, and an F1-score of 98.83%.

These updates highlight how EfficientNetB1 enhances text detection accuracy and robustness beyond both CNN-HR and heuristic-based methods.

Figure 9 presents some additional text detection results, such as character-level segmentations, demonstrating the model's ability to correctly recognize and extract text.

Figure 9: Real testing results Document text image detection results

5.4 Comparison with Existing Methods

We evaluate the performance of our suggested approach against that of some of the recent techniques in the literature ([6], [16], and [21]) using precision and recall, as presented in Table 3, to see whether it works. Recall that these algorithms were evaluated on

various data sets. For instance, [13] utilizes the publicly available AcTiV-D dataset, [21] utilizes Dataset 1 (HD), and [6] utilizes a custom-designed Arabic scene text dataset. We utilized a wide, internally designed dataset consisting of a diverse set of Arabic text images of real scenes for our model testing (CNN-HR and EfficientNetB1). Even though the datasets we used were different, our method fared better or equal, with EfficientNetB1 yielding the best precision and recall overall.

Table 3: Current work vs others

Work	Dataset Used	Precision	Recall
[21]	Dataset 1(HD)	81%	72%
[16]	AcTiV-D [22]	98%	82%%
[6]	Custom Arabic text dataset	97.20%	97.50%
This work CNN-HR	Our internal scene text dataset	97.35	98.92
This work EfficientNetB1	Our internal scene text dataset	98.72	98.93%

5.5 Final Observations

Experimental results indicate that the integration of CNN-HR verification and EfficientNetB1 is a robust and precise system for text detection. The suggested deep learning-based method significantly reduces the false detections in comparison to traditional heuristic-based methods and enhances overall performance. EfficientNetB1 has the best performance among all models discussed and is highly appropriate for real-world applications demanding precise text detection from natural images and document images. While several languages, such as Persian, Urdu, Kurdish, etc., use the Arabic script, this paper concerns itself with text in the Arabic language. Languages that have the same script are not differentiated by the system per se. Elaborate language identification is outside this paper and is a subject of interest for future work, although feature extraction would detect subtleties in the visual representation.

5. Conclusions

Two approaches to the design of an efficient Arabic text detection system were explored in this research:

1. CNN-HR Approach – Combination approach where CNN is used as a verification step, labeling regions identified either as non-text or text. With the CNN trained solely on text images, it removes the false positives that appear during heuristic detection and therefore improves accuracy.
2. EfficientNetB1 Method – A model that is deeper with the efficient extraction of features to lead the way towards successful text region detection and classification in images.

Key Findings and Contributions

- The validation process of CNN-HR enhances the overall accuracy from 88.31% to 96.46%.
- Accuracy is also enhanced by the EfficientNetB1 model to 97.77%, showing improved performance in the identification of Arabic text.
- The proposed framework performs outstandingly even in suboptimal conditions and detects the text bounding boxes correctly in diverse datasets and real-world settings.

Future Directions

- While significant progress has been made, further enhancements are needed to push Arabic text detection forward.
- Transformer-Based Models: Future work will explore transformer architectures to enhance accuracy and generalizability.

- Arabic Character Recognition: The system will be extended with a dedicated character recognition module to enable end-to-end text processing.
- Video Text Detection: Addressing the challenges of low-quality text extraction from videos remains a key focus for real-world applications.
- Advanced Feature Extraction: Techniques such as multi-scale feature fusion and contextual attention will be explored to improve the system's robustness in complex scenes, including handwritten or low-resolution text.

This study advances Arabic text detection by integrating state-of-the-art deep learning techniques, with promising applications in document analysis, real-time translation, and assistive technologies.

5. Disclosure and conflict of interest

The authors declare that they have no conflicts of interest.

References

[1] T. Fan, Q. Sun, J. Zhang, X. Tao, Y. Xu and T. He, "Text Detection in Sign Images Using Maximally Stable External Regions and Stroke Width Transform," 14th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), 2021.

[2] X. Gao, S. Han and C. Luo, "A detection and verification model based on SSD and encoder-decoder network for scene text detection," in IEEE Access, vol. 7, pp. 71299-71310, 2019.

[3] M. Mukhiddinov, "Scene Text Detection and Localization using Fully Convolutional Network," International Conference on Information Science and Communications Technologies (ICISCT), 2019.

[4] A. Amrouche, L. Falek, and H. Teffahi, "Design and Implementation of a Diacritic Arabic Text-To-Speech System," The International Arab Journal of Information Technology, Vol. 14, No. 4, July 2017.

[5] M. Fujitake, "A3S: Adversarial Learning of Semantic Representations for Scene-Text Spotting," International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2023.

[6] A. Amrouche, Y. Bentrcia, N. Hezil, A. Abed, K. N. Boubakeur and K. Ghribi, "Detection and Localization of Arabic Text in Natural Scene Images," First International Conference on Computer Communications and Intelligent Systems (I3CIS), Jijel, Algeria, 2022.

[7] S. Yousfi, S. A. Berrani and C. Garcia, "ALIF: A dataset for Arabic embedded text recognition in TV broadcast," 13th International Conference on Document Analysis and Recognition (ICDAR), 2015.

[8] J. Canny, "A Computational Approach to Edge Detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 6, pp. 679-698, Nov. 1986.

[9] M. Khan J., N. Said, A. Khan, N. Rehman and K. Khurshid, "Automated Latin Text Detection in Document Images and Natural Scene Images based on Connected Component Analysis," 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), 2019.

[10] E. A. Sekehravani, E. Babulak and M. Masoodi, "Implementing canny edge detection algorithm for noisy image," Bulletin of Electrical Engineering and Informatics. Vol. 9, No. 4, 2020, pp. 1404-1410.

[11] S. Liu, Y. Xian, H. Li and Z. Yu, "Text detection in natural scene images using morphological component analysis and Laplacian dictionary, " in IEEE/CAA Journal of Automatica Sinica, vol. 7, no. 1, pp. 214-222, 2020.

[12] W. Jia, L. Sun, Z. Zhong, X. Mo, G. Ma and Q. Huo, "A Robust Approach to Detecting Text from Images of Whiteboards and Handwritten Notes, "14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 2017.

[13] M. Moradi, S. Mozaffari, A. A. Orouji, "Farsi/Arabic text extraction from video images by

corner detection," 6th Iranian Conference on Machine Vision and Image Processing, 2010.

[14] W. Huang, Y. Qiao and X. Tang, "Robust Scene Text Detection with Convolution Neural Network Induced MSER Trees," European Conference on Computer Vision, 2014.

[15] D. Alsaleh and S. M. Larabi, "Arabic Text Classification Using Convolutional Neural Network and Genetic Algorithms," in IEEE Access, vol. 9, pp. 91670-91685, 2021.

[16] S. Manita, S. Mansouri, M. Zrigui and S.Berchech, "Arabic text detection in news video using RetinaNet," Procedia Computer Science, Vol 192, 2021, pp 796-803.

[17] Q. Ye, Q. Huang, W. Gao and D. Zhao, "Fast and robust text detection in images and video frames," Image and Vision Computing, 23(6):565–576, 2005.

[18] B. Benaziz, A. Zitouni and A. Amrouche, "Children Autism Detection Using Residual CNN and EfficientNetB1," 2023 International Conference on Electrical Engineering and Advanced Technology (ICEEAT), Batna, Algeria, 2023, pp. 1-5.

[19] J. Su, X.Yu, X. Wang, Z. Wang, G. Chao, "Enhanced transfer learning with data augmentation," Engineering Applications of Artificial Intelligence. Vol. 129, 2024.

[20] A. Amrouche, "DNN-based Arabic Printed Characters Classification, " 2nd International Conference on Electrical Engineering and Automatic Control (ICEEAC), Setif, Algeria, 2024.

[21] S. Mansouri, M. Charhad, M. Zrigui, "Arabic Text Detection in News Video Based on Line Segment Detector," Res. Comput. Sci. 132: 97-106, 2017.

[22] O.Zayene, M.Seuret, SM.Touj, J.Hennebert, N E. BenAmara, "Open Datasets and Tools for Arabic Text Detection and Recognition in News Video Frames", Proceeding of ICDAR, pp. 1-19, 2018.