Received: March 18, 2024. Revised: April 14, 2024.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Using Faster R-CNN to Detect and Recognize Arabic Handwritten Words

May Mowaffaq AL-Taee[1]* Sonia Ben Hassen Neji[1] Mondher Frikha[1]
Salah Taha Allawi[2]

[1]*École Nationale d'Électronique et de Télécommunications de Sfax, University of Sfax, Tunisia*
[2]*Department of Computer Science, College of Science, Mustansiriyah University, Baghdad, Iraq*
* Corresponding author's Email: may.tai@enetcom.u-sfax.tn

Abstract: Arabic handwriting recognition is a crucial area of computer vision research. Still, its complexity, diverse writing styles, and overlapping words have led to a lack of published research in this field. This paper suggests two new models to recognize handwritten Arabic words, depending on the Faster Region-Convolution Neural Network (Faster R-CNN). These models used two pre-trained networks during the feature extraction phase: The Visual Geometry Group-16 (VGG-16) network and the Residual Network (ResNet50) network. Models are independently trained and tested on two datasets: The Institut Für Nachrichtentechnik/Ecole Nationale d'Ingénieurs de Tunis (IFN/ENIT) dataset and the KFUPM Handwritten Arabic Text (KHATT) dataset. Test results showed that the proposed models give excellent results compared to others. The results of VGG16 and ResNet50 with the IFN/ENIT dataset reached accuracy rates of 92% and 100%, respectively. Meanwhile, the accuracy of the KHATT dataset reached 99.4% and 98% with VGG16 and ResNet50, respectively.

Keywords: Faster R-CNN, Handwritten words, Convolutional neural network, Feature extraction network.

1. Introduction

Recognition encompasses several fields, such as images, fingerprints, faces, numbers, and handwritten words. Typically, detecting handwritten words is categorized into two primary groups: online and offline recognition [1, 2]. While several works on offline Arabic handwriting recognition exist, most algorithms are limited to recognizing individual letters, numerals, or sentences with a restricted vocabulary. A few studies are available to identify unconstrained Arabic text with an extensive vocabulary. Recognizing Arabic text is challenging because of the various features of this language [3, 4]. These challenges include the complexity that arises because of the cursive nature of the script, the connectivity between characters, the diverse styles of writers, extensive vocabulary, the presence of ligatures, overlaps, and irregular spacing [1, 3]

The handwriting recognition system's principal goal is to convert handwritten text documents from digital image format into encoded character format documents so that programs for word processing can read and change them. [5]. Handwriting recognition is used in various fields, including postal code recognition, office automation, writer identity, document processing, signature verification, and automated check processing in banks [5-7]. For the past 20 years, researchers have used a variety of tools to study Arabic handwriting recognition in depth, including Support Vector Machines (SVM), Hidden Markov Models (HMM), Convolutional Neural Networks (CNN), Deep Neural Networks (DNN), Multilayer Perception (MLP), Recurrent Neural Networks (RNN), and more [8].

CNNs are a promising iteration of DNN that has shown outstanding results in applications like image categorization, particularly in handwritten character recognition [9]. Faster R-CNN [10] as a combination of Region Proposal Networks (RPNs) and Fast R-CNN [11] works well in object detection. It achieves end-to-end training specifically for the task using novel RPNs and is further promoted based on region proposal methods and Fast R-CNN [9].

Recently, researchers have endeavoured to develop methods for recognizing Arabic handwritten words using the IFN/ENIT. In 2021, Maalej and Kherallah [12] suggested an offline Arabic handwriting recognizer that used a Multi-Dimensional Long Short-Term Memory Network (MDLSTM) and Rectified Linear Units (ReLUs) to fix issues with vanishing gradients and dropout to avoid overfitting. Evaluated on the IFN/ENIT database, the systems achieved a label error rate of 11.40%. Moreover, in 2022, Ali and Mallaiah [13] proposed a model for Arabic handwriting recognition using CNN and Support Vector Machine (SVM). They applied dropout to the model and showed its efficiency in many Arabic scripts. The model was tested using the IFN/ENIT, HACDB, AHDB, and AHCD databases. The test results using IFN/ENIT datasets showed that the proposed model with dropout achieved 98.58%, while the model without dropout achieved 96.50%. More recently, in 2023, Gader et al. [7] developed a model with three components: CNN, RNN (ConvLSTM) Convolutional Long Short-Term Memory, and Connectionist temporal classification (CTC). CNN was used for feature extraction, while RNN was used for spatiotemporal prediction. Moreover, the CTC was applied to infer information from the input image. The model was tested using different scenarios from the IFN/ENIT dataset, such as training the model with groups (a, b, c), testing with group (d), and so on with the rest of the groups. The recognition rate is approximately 99.01%, 95.05%, and 96.57% for abc-d, abcd-e, and abcde-f, respectively.

In contrast, other researchers have endeavoured to develop methods for recognizing Arabic handwritten words using the KHATT datasets. In 2019, Sana et al. [14] introduced a new method for recognizing out-of-vocabulary words as sub-word units. They tested this approach on two handwriting recognition methods: one based on customized HOG features and a Bidirectional Long Short-Term Memory BLSTM and CTC, and the other on CNN's learned features and the MDLSTM as a classifier with CTC. The experiments were performed using the KHATT dataset. The authors found that combining full-word models of character, morpheme, and PAWs was successful, especially when using the CNN-MDLSTM. Sub-word and character language models ensured significant coverage of OOVs. The combination of the Full-Word, Morphemes, Paws, and Character models achieved a WER of 20.86%. In 2020, Riaz et al. [15] Enhanced their prior approach, which was based on MDLSTM, with the CTC layer as the last layer, Through the application of five data augmentation techniques and a deep learning strategy. They achieved an accuracy rate of 80.02% for character recognition, a character error rate of 4.22%, and a label error rate of 19.98% in the KHATT dataset. In addition, in 2020, Zouhaira et al. [3] Suggested open-vocabulary offline recognition method for handwritten Arabic text was developed, focusing on enhancing image quality through preprocessing steps. The approach used a deep Convolutional Recurrent Neural Network (CRNN) model with a VGGNet architecture and a BLSTM layer with CTC beam search decoder. Also the Dropout regularization technique is applied. The model achieved significant results, with an accuracy of 87.39% and a Word Error Rate (WER) of 12.61% when evaluated on the KHATT databases.

Researchers have endeavored to devise techniques for Arabic word recognition using two datasets: the IFN/ENIT and KHATT datasets. In 2022, Gader and Echi [6] The study introduced a deep learning approach for extracting handwritten Arabic words, employing an Attention-based CNN-ConvLSTM model alongside a CTC function. The method showed powerful performance across the KHATT, IFN/ENIT, and AHDB datasets, achieving an extraction rate of 91.7% on the KHATT database and 94.1% on the IFN/ENIT database. In 2023, Lamtougui et al. [1] Propose a novel approach incorporating a CNN, a BLSTM, and a CTC layer with a Word Beam Search (WBS) decoder. A data augmentation technique was employed to enhance data quality during training. The model was trained and evaluated using two databases, KHATT and IFN/ENIT, resulting in an accuracy of 92.11% for IFN/ENIT and 80.15% for KHATT.

While effective, these techniques possess several limitations. MDLSTM, for instance, is susceptible to the vanishing gradient problem. The authors in [12] incorporated the Rectified Linear Unit (ReLU) to address this challenge. To reduce overfitting-related issues, [3, 12, 13] used regularization techniques like dropout. During the feature extraction step, [1, 3] employed a sliding window approach to feeding data into the CNN, which involves scanning the original image horizontally using a sliding window that moves from right to left. The dimensions of the window matched the width of the input text-line images. Several of these methods required substantial data volumes. Consequently, [1] implemented data augmentation three times on individual lines of text, while [15] augmented the data five times for each single line of text. The authors in [1] used a method of CNN, BLSTM, and CTC, coupled with a Word Beam Search (WBS) decoder, to decode extensive training samples. Similarly, [14] employed various sub-word-based language models. Some methods [3,

14, 15] used an extra preprocessing stage, including rectifying line skew. These methodologies used large training datasets, therefore escalating computational costs and augmenting memory consumption.

This study aims to use the characteristics of Faster R-CNN in object recognition, where the Faster R-CNN algorithm marks a notable advancement in object detection, seamlessly integrating the Region Proposal Network (RPN) and Fast R-CNN into a unified network architecture. This amalgamation leads to:

1. Improved accuracy and real-time performance. By leveraging the capabilities of region proposal networks (RPNs), the algorithm overcomes the long computation time associated with earlier methods, such as selective search (SS).
2. Efficiently extracting region proposals tailored to input samples, employing anchor boxes to precisely identify regions of interest within an image, and facilitating effective object localization and classification.
3. RPN sharing full-image convolutional features with the detection network. This lets it make almost cost-free region suggestions and speeds up object detection. This streamlined approach mitigates the computational bottleneck typically associated with traditional object detection methods, significantly reducing processing time.

In contrast, pre-training a backbone network yields several advantages. It accelerates the training process, conserves computational resources, and provides a robust initialization point, enhancing feature extraction capabilities. Additionally, leveraging a pre-trained network helps address challenges related to limited labeled data, promoting faster convergence during training and ultimately improving accuracy and performance.

The significant contributions of this paper include:

1. The proposed models are considered the first to use the Faster R-CNN approach with the IFN/ENIT and KHATT datasets.
2. A Faster R-CNN algorithm was implemented based on pre-trained models VGG16 and ResNet50, which helps improve accuracy and performance and reduces processing time.
3. Using Soft Non-Maximum Suppression (Soft-NMS) in the last step instead of NMS solves multiple detections of the same object in an image. It increases the number of detections and their accuracy.
4. We manually created the bounding box annotations since the IFN/ENIT and KHATT datasets lacked them. These annotations were crucial for the training process, reducing the range of searches for object features and the time needed for searches.
5. Using less data than previous models and getting excellent results.

This paper includes Section 2, which discusses the components of the suggested method. Section 3 displays the steps for implementing the proposed method. Section 4 displays the experimental results. Finally, Section 5 summarizes the conclusions and future work to develop the models.

2. Methodology

Our approach uses the Faster R-CNN [10] and two pre-trained networks, VGG16 [16] and ResNet50 [17], to detect and recognize handwritten Arabic words in the IFN/ENIT [18] and KHATT [19] datasets. This section will discuss the materials used in the proposed method.

2.1 IFN/ENIT dataset

The IFN/ENIT database contains 32492 images of Arabic words written by over 1000 writers. These words represent the names of 937 villages and towns in Tunisia and include 5 groups (a, b, c, d, and e). Many research groups have used this dataset, which is one of the most common databases for handwritten Arabic text recognition research [1, 6, 20]. Fig. 1 displays some examples of images of handwritten Arabic words from the IFN/ENIT dataset. Table 1 displays statistics for the number of words in each group.

2.2 KHATT dataset

The KHATT database was presented at the 13th International Conference on Frontiers in Handwriting Recognition (ICFHR) in 2012, intending to facilitate research in character recognition regarding Arabic script. This dataset contains unrestricted writing styles. This data was written by 1000 writers from different countries, age groups, genders, and education levels and consists of 4000 paragraphs;

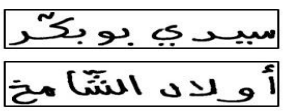


Figure.1 Examples of images of handwritten Arabic words from the IFN/ENIT dataset.

Table 1. Statistic for number word in each group

Group name	a	b	c	d	e
Words	6537	6710	6477	6735	6033
Total words	32492				

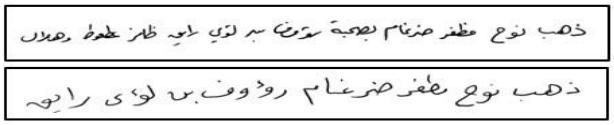


Figure. 2 Examples of images of handwritten Arabic text lines from the KHATT dataset.

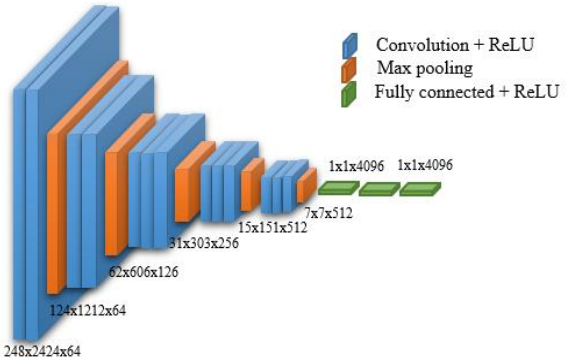


Figure.3 The VGG16 network architecture

it is divided into 2000 unique, randomly selected paragraphs with different text contents and 2000 fixed [6, 21, 22]. Fig. 2 displays some examples of images of handwritten Arabic text lines from the KHATT dataset.

2.3 VGG16 network

The VGG16 [16] network has 13 convolution layers activated by Rectified Linear Units (ReLU) and three fully connected layers. It also has four pooling levels. The last fully linked layer is eliminated, leaving only the front portion of the convolutional layer, which forms the network core [23, 24]. Fig. 3. shows the general structure of the VGG16 network.

2.4 ResNet50 network

The ResNet50 [17] architecture comprises two modules: convolution and identity blocks. Because the convolution block's input and output dimensions differ, they cannot be linked in series. Therefore, the network's dimension should be changed. The input dimension of the identity block is the same as the output dimension, which may be connected in series and used to deepen the network [25, 26]. Fig. 4. illustrate the general structure of the ResNet50 network.

2.5 Faster R-CNN

Faster R-CNN [10] includes feature extraction, Region Proposal Network, and the Fast R-CNN method (detector). Fig. 5 illustrate the overall structure of the Faster R-CNN.

2.5.1 Feature extraction

The feature extraction step is crucial to the overall performance of the Faster R-CNN algorithm. This stage uses CNN, a leading-edge object detection technique that employs a set of convolution and pooling operations to extract essential features from images. Each image and its corresponding annotations are fed to the ResNet50 or the VGG16 pre-trained networks to ensure efficient and effective image feature extraction [27].

2.5.2 Region proposal network

The RPN is a Fully Convolutional Network (FCN) that generates exact regional proposals using shared full-image convolutional features. It uses a 3×3 sliding window approach to process input and create a feature vector, with 9 anchors generated at each image point with three aspect ratios (1:1, 2:1, 1:2) and three scales (32, 64, and 128) in the center. Two fully connected layers process proposals to determine the likelihood of an object being present in the proposed window. One layer predicts the object's bounding box coordinates, while the other determines if the proposal is an object (a word) or a background.

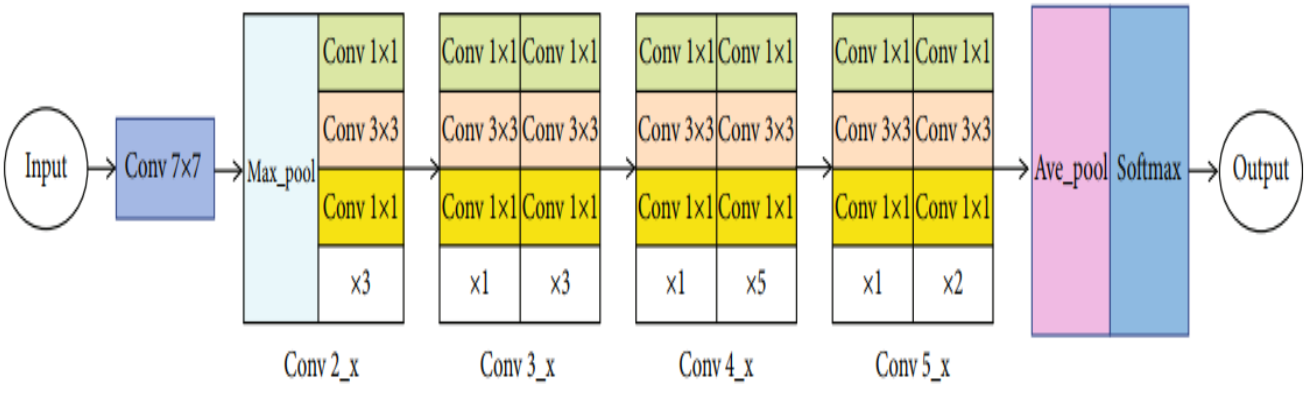


Figure.4 The resNet50 network architecture

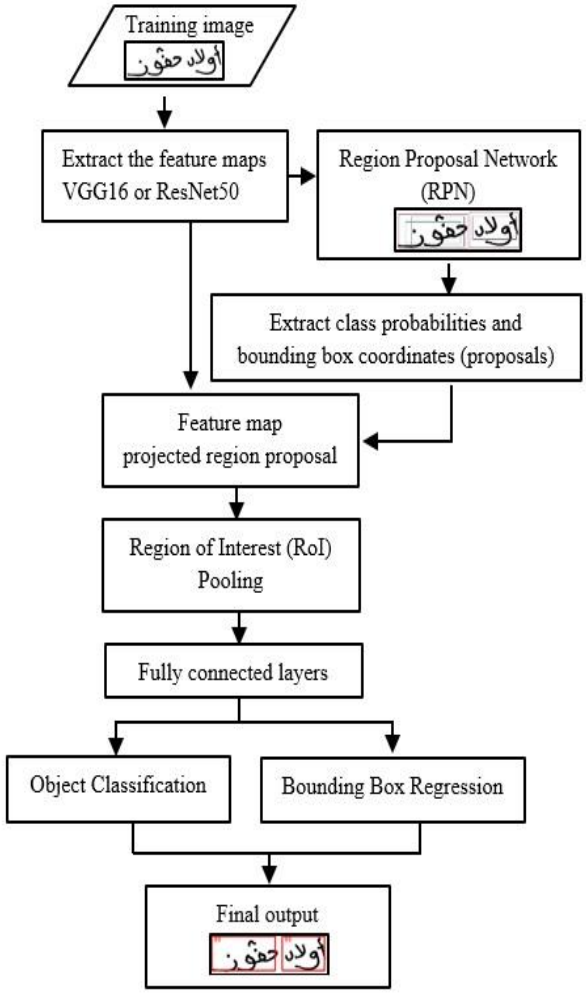


Figure.5 The general structure for the Faster R-CNN

The RPN can be trained from end to end and feeds these proposals into the Fast R-CNN for detection [27]. *IoU*, a crucial indicator in object identification, is the best method to determine the distance between the ground truth box and the predicted bounding box in the regression task. Eq. (1) is used to calculate the *IoU*.

$$IoU = \frac{Anchor \cap GTBox}{Anchor \cup GTBox} \tag{1}$$

Where: The *IoU* represents the ratio of the intersection area and the union area of the ground truth bounding box (*GTBox*) with the *Anchor*. *Anchors* are suggested outputs assigned an objectness score determined by (*IoU*) score [28].

The RPN uses two types of anchors: positive and negative. The positive anchor is assigned when the *IoU* score for any ground truth box exceeds 0.7, whereas the negative anchor is assigned when it is below 0.3. Anchors neglected do not influence the training loss, where their degrees range from 0.3 to 0.7; in contrast, the subsequent network module is trained using the remaining positive and negative anchors [29]. Eq. (2) determines whether the anchors are negative or positive based on the threshold value.

$$p^* = \begin{cases} -1 & if \quad IoU \ < 0.3 \\ 1 & if \quad IoU \ > 0.7 \end{cases} \tag{2}$$

Eq. (3) is used to calculate the loss function of the whole network:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{i}{N_{reg}} \sum_i p_i^* \, L_{reg}\,(t_i, t_i^*) \tag{3}$$

Where i is the index of anchors, p_i is the probability that the i-th anchor is predicted to be the true label. In contrast, p_i* is the presence or absence of a target for the anchor. At the same time, t_i is the prediction of the bounding box regression parameter of the i-th anchor. In addition, t_i* is the ground truth box corresponding to the i-th anchor, N_{cls} is the batch size, N_{reg} is the number of anchor positions, and λ is the balance parameter. While L_{cls} is a binary log loss, and L_{reg} is a smoothed L1 loss.

Back-propagation with stochastic gradient descent (SGD) can be used to train a Faster R-CNN from beginning to end, which helps improve the loss function [10], [27].

For object detection models that decrease RPN proposal redundancy, NMS is crucial. Choosing the detection box with the highest classification score and removing boxes with considerable overlap decreases recommendations while maintaining detection accuracy [30]. In contrast, Soft-NMS is used instead of the NMS after the classification stage to deal with multiple detections of the same class in an image, improve localization accuracy, deal with overlapping detections, and let the bounding box selection and confidence score fine-tuning happen [31].

2.5.3 Fast R-CNN detector

A detection network receives the feature map and the regions of interest generated by the previous networks as input. Then, the RoI pooling selects a specific area from the feature map and resizes it to a fixed size. After processing the feature maps and proposals, the information is aggregated and used to generate proposal feature maps of fixed sizes. These maps are then transformed into vectors and input into fully connected layers [9]. The classification and regression layer comprises a fully connected layer that displays the class assigned to each word. The bounding box regression generates a bbox that shows

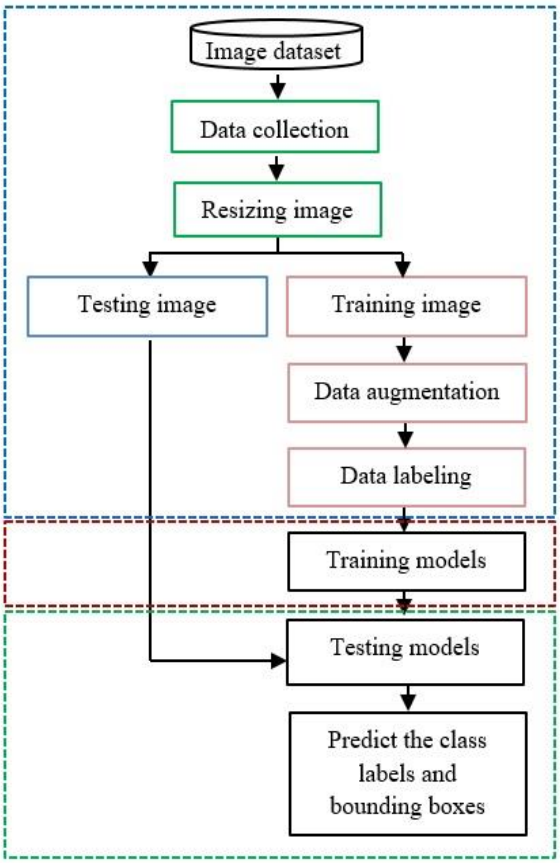


Figure.6 The general outline for the proposed models

the last position of the recognized word [27, 28, 32, 33].

3. Implementation models

This work is divided into three major stages: data preparation, training, and testing the models. Fig. 6 displays the outline of the proposed work.

3.1 Data preparation

3.1.1 Data collection

Our work uses 16 classes from the KHATT dataset and 20 from the IFN/ENIT dataset. Table 2 illustrates the names of the classes used in our work from the IFN/ENIT and KHATT datasets.

3.1.2 Resizing image

Because of the different image sizes in the IFN/ENIT dataset, the images were resized to a fixed size without distortion by performing several steps.

1. All-white areas were removed from the images.
2. The size of the images was changed to 256*64.
3. Add a white area with 6 pixels on each side of the image to facilitate the labeling process around each class in the image, resulting in a final image with a size of 262×72 pixels used during the training and testing phases.

Table. 2 Class names in the IFN/ENIT and KHATT datasets.

IFN/ENIT dataset		KHATT dataset	
English	Arabic	English	Arabic
Awlad	أولاد	Thahb	ذهب
Hafooz	حفوز	Nooh	نوح
Alshamikh	الشامخ	Mathfar	مظفر
Bie'r	بئر	Dhirgham	ضرغام
Marwa	مروة	Bisuhbat	بصحبة
Dawar	دوار	Ra'aoof	رؤوف
Alliwata	اللواته	Bin	بن
Hay	حي	Loa'y	لؤي
Alsalah	الصلاح	Raayq	رايق
Ra's	رأس	Thfir	ظافر
Althiraa'	الذراع	Ata'oot	عطعوط
Sabat	سبعة	Wa	و
Abaar	أبار	Hilal	هلال
Tel	تل	khazin	خازن
Alghizlan	الغزلان	Afeef	عفيف
Rabaya'	ربايع	Lilhij	للحج
Seedi	سيدي		
Dhahir	ظاهر		
Boo	بو		
Bakir	بكر		

In contrast, because the images in the KHATT dataset are of different sizes and contain non-uniform white areas, we took several steps to standardize their size.

1. Identify and remove white areas from images, calculate the width and height values for all images, and then find the maximum width and height value.
2. Calculating a new size by adding 5 pixels to the maximum width and height value to make placing tags around each category in the image easier.
3. Create a white image using the new size values calculated in step 2. Then, create a new image by adding the image resulting from step 1 in the middle of the white image.

3.1.3 Data splitting

The data was divided into 80% for training and 20% for testing. The total number of images used in each database was 1000, divided into 800 images for training and 200 for testing.

3.1.4 Data augmentation

A data augmentation approach was applied to solve the problem of data imbalance in the IFN/ENIT dataset during the training phase, using three data augmentation techniques: variance, expansion, and erosion. After augmenting, the total number of

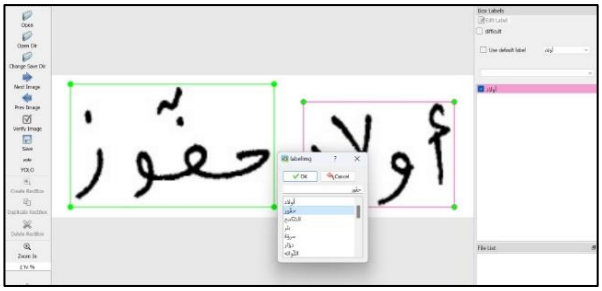


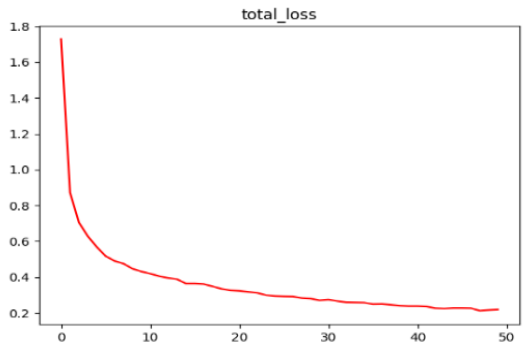
Figure.7 Creating the bounding box manually for the classes in the image.

images in the IFN/ENIT dataset was 3200 in the training phase.

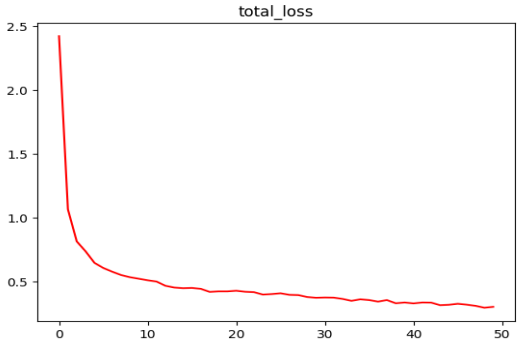
3.1.5 Data labeling

The LabelImg tool [34] was used to create the bounding box (bbox) annotations manually around each class in the image used in the training and testing phases. The bbox annotations contain each class's name and bbox values (xmax, xmin, ymax, ymin, height, and width). Each image has its own XML file, and then the XML files are grouped into one CSV file and then converted to a TXT file used in the training phase. Fig. 7 illustrates an example of manually creating the bounding box for the classes in the image.

3.2 Training models

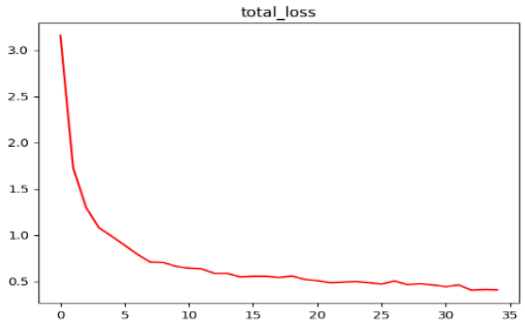


(a)

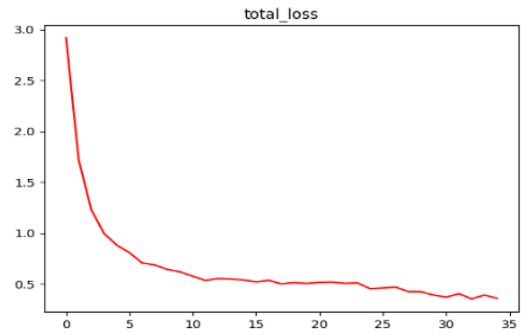


(b)

Figure. 8 The total loss for IFN/ENIT dataset: (a) VGG16 and (b) ResNet50



(a)



(b)

Figure. 9 The total loss for KHATT dataset: (a) VGG16 and (b) ResNet50

The training phase was performed separately for each dataset. The IFN/ENIT dataset was trained with 50 epochs and 3200 iterations: the VGG16 model required 46 hours, while the ResNet50 model required only 23 hours. Fig. 8 illustrates the total loss for VGG16 and ResNet50 after training the model with 50 epochs.

The KHATT dataset was trained with 35 epochs and 800 iterations; the VGG16 model requires 76 hours, while the ResNet50 model requires only 41 hours. Fig. 9 illustrates the total loss for VGG16 and ResNet50 after training the model with 35 epochs.

The learning rate was 1e-5, and we changed the RPN setup by modifying the three scales to (32, 64, 128) to enhance the precision of detecting small-sized objects while maintaining aspect ratios of (1:1, 2:1, and 1:2). The code written in Python was executed on an NVIDIA Processor Core i9. The entire training is done on a CPU.

3.3 Test models

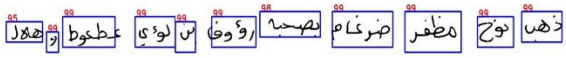
The testing process is performed on each dataset independently. The IFN/ENIT dataset was tested using 200 images. Fig. 10 illustrates an example of the result from the test phase.

The KHATT dataset used 200 images in the test phase. Fig. 11 illustrates an example of the result from the test phase conducted on the KHATT dataset.

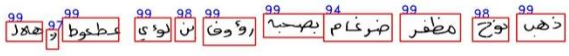
(a)

(b)

Figure.10 Example of the test result of the IFN/ENIT dataset: (a) VGG16 result and (b) ResNet50 result



(a)



(b)

Figure.11 Example of the test result of the KHATT dataset: (a) VGG16 result and (b) ResNet50 result

4. Result and discussion

To evaluate the efficiency of the proposed models, we use several evaluation metrics, including Recall, F1_score, Accuracy, and Precision. These metrics are defined as follows: [6, 33].

$$Recall = \frac{TP}{TP+FN} \tag{4}$$

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP} \tag{5}$$

$$F1\_Score = \frac{2*Precision*Recall}{Precision+Recall} \tag{6}$$

$$Precision = \frac{TP}{TP+FP} \tag{7}$$

Where *TN*, *FP*, *TP*, and *FN* stand for True Negative, False Positive, True Positive, and False Negative, respectively. These results depend on a crucial parameter, the *IoU* threshold, which determines whether a predicted bounding box is a False Positive or a True Positive. Moreover, we use the mean average precision *mAP*, an essential metric in target detection, to evaluate the proposed models. It is defined in Eq. (8) as:

$$mAP = \frac{1}{M} \sum_{q=1}^{M} AP_q \tag{8}$$

Where: AP_q is the average precision of the qth class and M is the total number of classes.

The best result was achieved by testing the models on the IFN/ENIT dataset after 25 epochs with VGG16, while the best was with ResNet50 after 40 epochs. Table 3 shows the testing results of the accuracy and *mAP* for the best epoch in each model with the IFN/ENIT dataset.

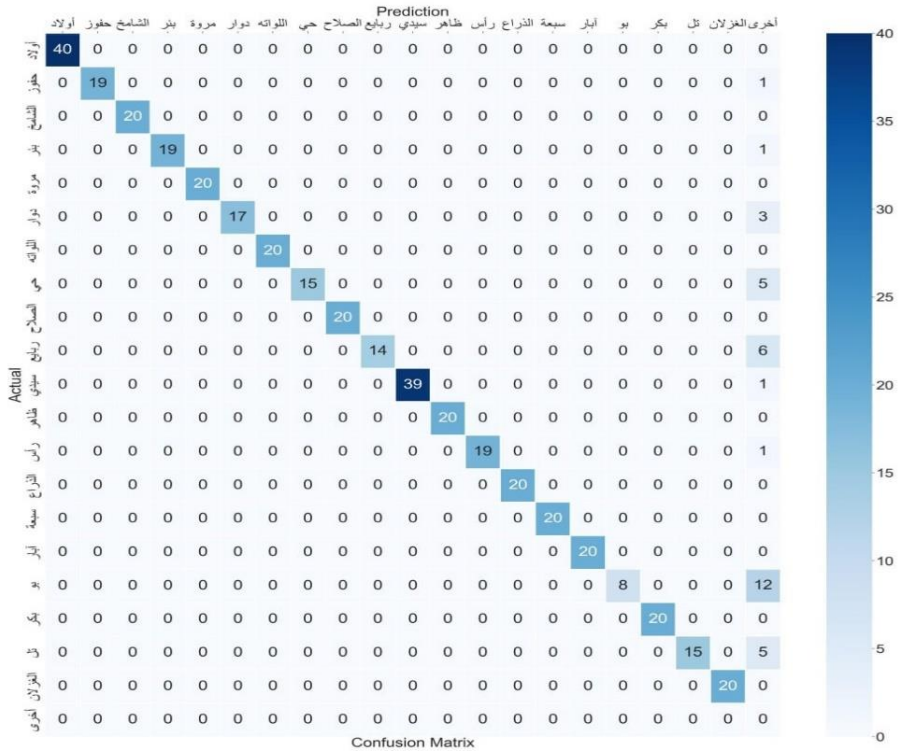


Figure. 12 Confusion matrix for the IFN/ENIT dataset with VGG16

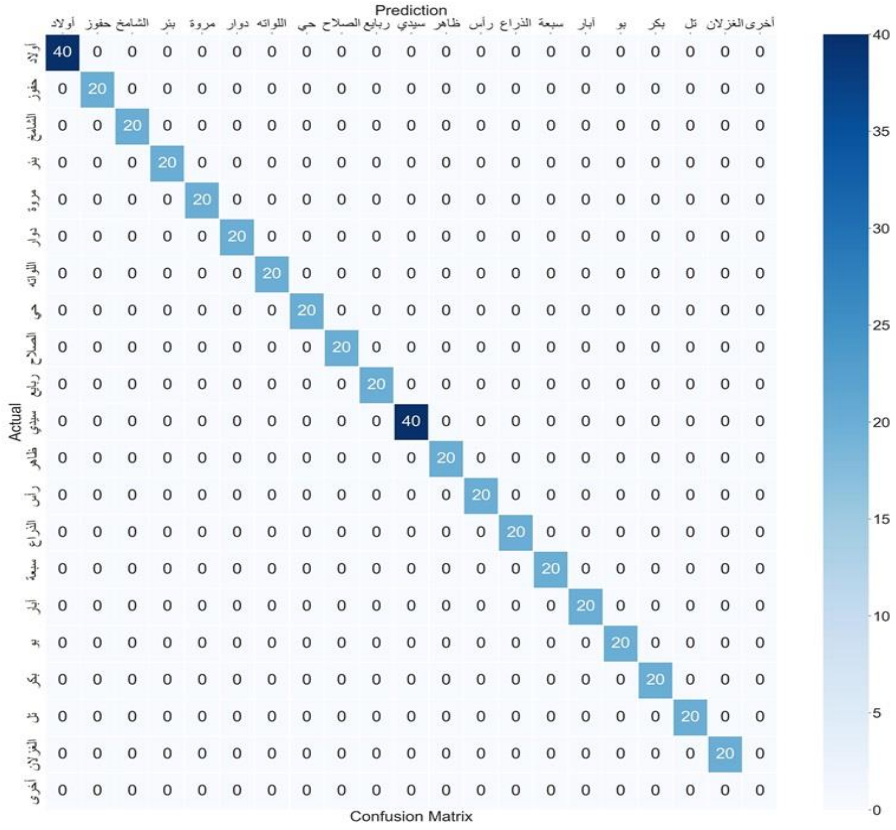


Figure. 13 Confusion matrix for the IFN/ENIT dataset with ResNet50

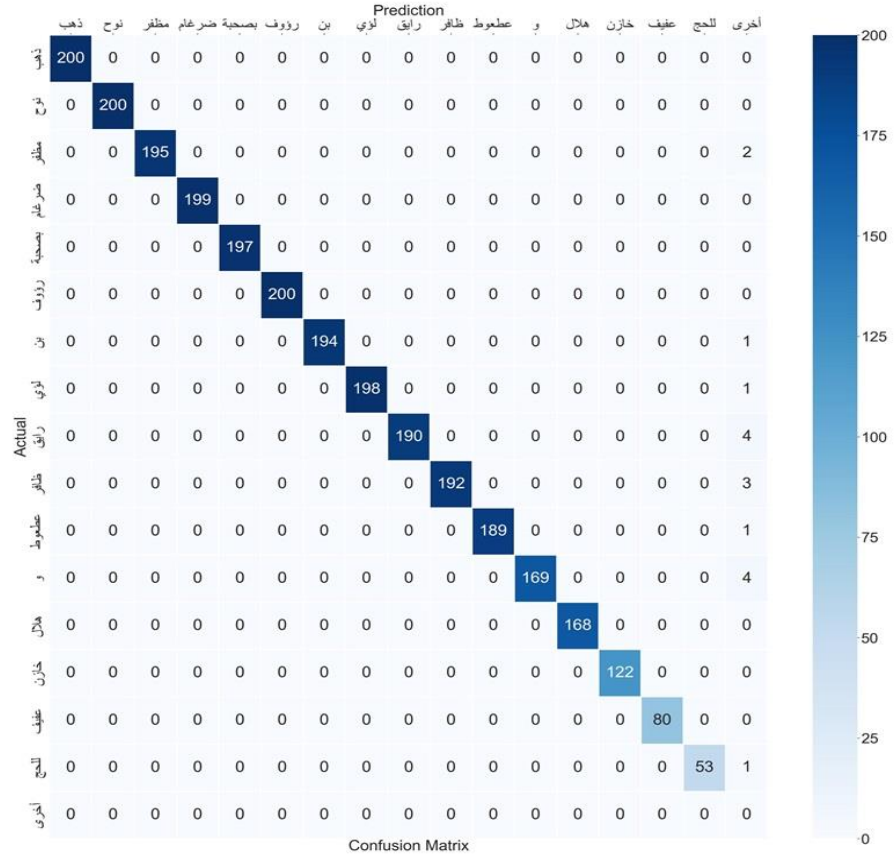


Figure. 14 Confusion matrix for the KHATT dataset with VGG16

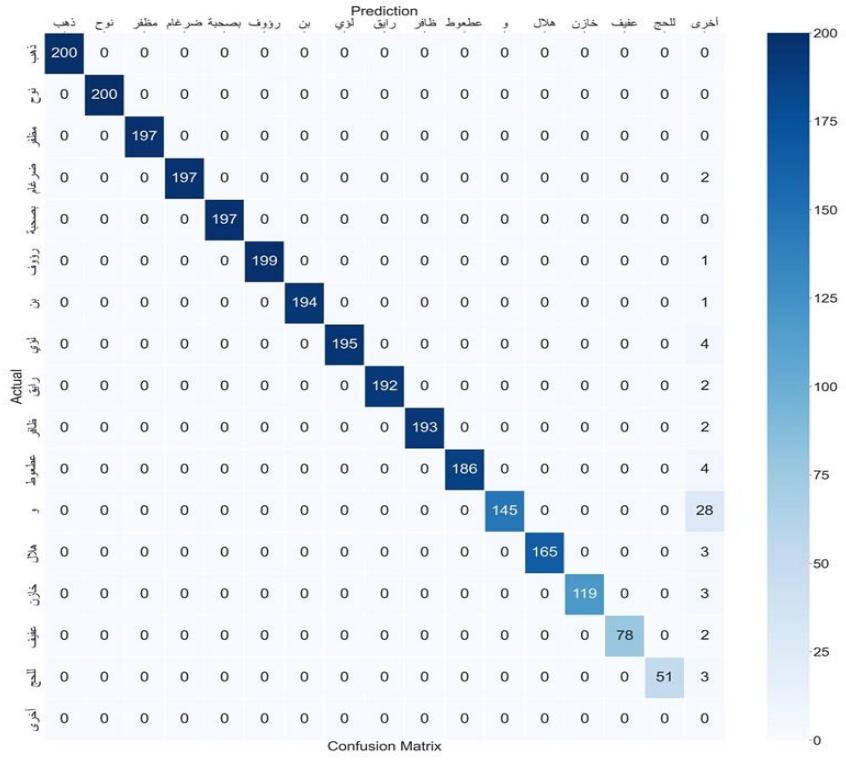


Figure. 15 Confusion matrix for the KHATT dataset with ResNet50

Table 3. Test results to the IFN/ENIT dataset

Model	Best Epoch	Accuracy	mAP
VGG16	25	92%	99.3%
ResNet50	40	100%	99.4%

Table 4. Test results to the KHATT dataset

Model	Best Epoch	Accuracy	mAP
VGG16	35	99.4%	99.9%
ResNet50	35	98%	99.8

Table 5. Accuracy after applying different values of IoU threshold

Dataset	Model	IoU value		
		0.5	0.45	0.4
KHATT	VGG16	98.4 %	99.1 %	99.4 %
	ResNet50	97.3 %	98 %	98 %
IFN/ENIT	VGG16	91.4%	91.8%	92%
	ResNet50	100%	100%	100%

Fig. 12 shows the confusion matrix for the IFN/ENT dataset results for the best epoch with VGG16. In contrast, Fig. 13 shows the confusion matrix for the IFN/ENT dataset results for the best epoch with ResNet50.

Similarly, testing the models on the KHATT dataset after 35 epochs with VGG16 and ResNet50 yielded the best results. Table 4 shows the testing results of the accuracy and *mAP* for the best epoch in each model with the KHATT dataset. Fig. 14 shows the confusion matrix for the KHATT dataset results for the best epoch with VGG16. At the same time, Fig. 15 shows the confusion matrix for the KHATT dataset results for the best epoch with the ResNet50 network.

Determining the appropriate *IoU* threshold value depends on the type and purpose of the application. In this work, three values of the *IoU* threshold (0.4, 0.45, and 0.5) were tested, and the results obtained showed that the value (0.4) gives the best result. Table 5 shows the accuracy results after applying different values of the *IoU* threshold with the best epochs.

Table 6 compares our suggested models to other methods that used the IFN/ENIT and KHATT datasets.

Other models have some limitations despite their efficiency. They do require large training samples, which drive up computational expenses. Furthermore, some approaches significantly increase computational complexity, while other models use various regularization techniques to avoid overfitting.

Because our suggested models do not need large training samples or a combination of language do not require the integration of organizational techniques, dictionaries, or linguistic models. Despite these advantages, manual labeling or annotation of the

Table 6. Comparison the proposed models with other models by using IFN/ENIT and KHATT datasets.

Ref.	Model	Dataset	Training - Testing	Accuracy
[7]	CNN, ConvLSTM, CTC	IFN/ENIT	abc-d abcd-e abcde-f	99.01% 95.05% 96.57%
[12]	MDLSTM, Dropout, ReLUs	IFN/ENIT	abc-de	Label error 11.40%
[13]	CNN based SVM	IFN/ENIT	-	with dropout 98.58% without dropout 96.50%
[3]	CNN, BLSTM, CTC	KHATT	9475 lines - 2007 lines	87.39%
[14]	CNN, MDLSTM, CTC	KHATT	-	79.14%
[15]	MDLSTM, CTC	KHATT	24125 lines - 4685 lines	80.02%
[1]	CNN, BLSTM, CTC, WBS	IFN/ENIT	abcd-e	92.11%
		KHATT	14475 lines - 2898 lines	80.15%
[6]	CNN, Attention convLSTM, CTC	IFN/ENIT	4800 lines - 200 images	94.1%,
		KHATT		91.7%
Our method	Faster R-CNN VGG16, ResNet50	IFN/ENIT	3200 images - 200 images	92% 100%
	Faster R-CNN VGG16, ResNet50	KHATT	800 lines - 200 lines	99.4% 98%

training images is relatively complex.

5. Conclusion

With a DL approach based on the Faster R-CNN architecture, the current work is the first attempt to localize and recognize handwritten Arabic words in the IFN/ENIT and KHATT datasets. This work aims to provide new, more efficient methods for raising the accuracy of handwriting recognition. The testing results show that the models effectively localize and recognize handwritten Arabic words despite the few training samples. The results showed the ResNet50 model, trained on the IFN/ENIT dataset, attained a perfect accuracy of 100%, whereas the VGG16 model achieved an accuracy of 92%. By comparison, the VGG16 model using the KHATT dataset had the highest accuracy of 99.4%, whereas the ResNet50 model reached an accuracy of 98%. The results also showed that using a suitable *IoU* value increases the number of discoveries and the accuracy of the models. Using Faster R-CNN, the computational cost is low because it relies on a few training samples and lacks reliance on organizational strategies or linguistic models. In our future endeavors, our primary aim will be to enhance existing work by using alternatively trained networks. This will enable us to achieve higher levels of accuracy in our findings. We will also concentrate on constructing a character recognition model based on deep learning.

Conflicts of Interest

The authors declare no conflicts of interest.

Author Contributions

Conceptualization, MMA, SBHN, and MF; methodology, NMA, and STA; software, MMA; validation, MMA, SBHN, MF, and STA; formal analysis, MMA, and STA; investigation, MMA, and STA; resources, MMA, and STA; data curation, MMA, and STA; writing—original draft preparation, MMA, and STA; writing—review and editing, SBHN, and MF; visualization, MMA; supervision, SBHN, and MF; project administration, not found; funding acquisition, not found.

Acknowledgements

The authors express gratitude for the assistance from the National School of Electronics and Communications in Sfax.

References

[1] H. Lamtougui, H. El Moubtahij, H. Fouadi, and K. Satori, “An Efficient Hybrid Model for Arabic Text Recognition”, *Computers Materials Continua*, Vol. 74, No. 2, pp. 2871-2888, 2023.

[2] M. M. Al-Taee, S. B. H. Neji, and M. Frikha, “Handwritten Recognition: A survey”, In: *Proc. of IEEE 4th International Conf. Image*

Processing Appllications and Systsems, Genova, Italy, pp. 199-205, 2020.

[3] Z. Noubigh, A. Mezghani, and M. Kherallah, "Open Vocabulary Recognition of Offline Arabic Handwriting Text Based on Deep Learning", In: *Proc. of 20th International Conference on Intelligent Systems Design and Applications*, pp. 92-106. 2021.

[4] S. Hamida, O. El Gannour, B. Cherradi, H. Ouajji, and A. Raihani, "Efficient feature descriptor selection for improved Arabic handwritten words recognition", *International Journal of Electrical and Computer Engineering*, Vol. 12, No. 5, pp. 5304-5312, 2022.

[5] M. Eltay, A. Zidouri, and I. Ahmad, "Exploring Deep Learning Approaches to Recognize Handwritten Arabic Texts", *IEEE Access*, Vol. 8, pp. 89882-89898, 2020.

[6] T. B. A. Gader and A. K. Echi, "Attention-based CNN-ConvLSTM for Handwritten Arabic Word Extraction", *Electronic Letters on Computer Vision and Image Analysis*, Vol. 21, No. 1, pp. 121-134, 2022.

[7] T. Gader, I. Chibani, and A. Echi, "Arabic Handwriting off-Line Recognition Using ConvLSTM-CTC", In: *Proc. of 12th international Conf. on Pattern Recognition Applications and Methods*, Lisbon, Portugal, pp. 529-533, 2023.

[8] M. Elleuch, S. Jraba, and M. Kherallah, "The Effectiveness of Transfer Learning for Arabic Handwriting Recognition using Deep CNN", *Journal of Informaton Assurance and Security*, Vol. 8, pp. 85-93, 2021.

[9] J. Yang, P. Ren, and X. Kong, "Handwriting Text Recognition Based on Faster R-CNN", In: *Proc. of Chinese Automation Congress CAC 2019*, pp. 2450-2454, 2019.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, No. 6, pp. 1137-1149, 2016.

[11] R. Girshick, "Fast R-CNN", In: *Proc. of the IEEE International Conf. on Computer Vision*, pp. 1440-1448, 2015.

[12] R. Maalej, and M. Kherallah, "ReLU to Enhance MDLSTM for Offline Arabic Handwriting Recognition", In: *Proc. of 19th International Conference on Intelligent Systems Design and Applications (ISDA 2019)*, Springer, Cham, pp. 386-395, 2021.

[13] A. A. A. Ali, and S. Mallaiah, "Intelligent handwritten recognition using hybrid CNN architectures based-SVM classifier with dropout", *Journal King Saud University - Computer and Information Sciences*, Vol. 34, No. 6, pp. 3294-3300, 2022.

[14] S. K. Jemni, Y. Kessentini, and S. Kanoun, "Improving Recurrent Neural Networks for Offline Arabic Handwriting Recognition by Combining Different Language Models", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 34, No. 12, p. 2052007, 2020.

[15] R. Ahmad, S. Naz, M. Afzal, S. Rashid, M. Liwicki, and A. Dengel, "A Deep Learning Based Arabic Script Recognition System: Benchmark on KHAT", *The International Arab Journal of Information Technolog*, Vol. 17, No. 3, pp. 299-305, 2020.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", In: *Proc. of 3rd International Conf. Learning Representations ICLR*, San Diego, USA, pp. 1-14, 2015.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, USA, pp. 770-778, 2016.

[18] M. Pechwitz, S. S. Maddouri, V. Märgner, N. Ellouze, and H. Amiri, "IFN / ENIT-database of handwritten Arabic words", In: *Francophone International Conf. on writing and Document", CIFED'02, Hammamet, Tunisia*, pp. 127-136, 2002.

[19] S. A. Mahmoud, I. Ahmad, M. Alshayeb, W. G. Al-Khatib, M. T. Parvez, G. A. Fink, V. Märgner, and H. El-Abed, "KHATT: Arabic offline Handwritten Text Database", In: *Proc. of 2012 International Conf. on Frontiers in Handwriting Recognition, IWFHR*, Bari, Italy, pp. 449-454, 2012.

[20] S. Haboubi, T. Guesmi, B. M. Alshammari, K. Alqunum, A. S. Alshammari, H. Alsaif, and H. Amiri, "Improving CNN-BGRU Hybrid Network for Arabic Handwritten Text Recognition", *Computers Materials & Continua*, Vol. 73, No. 3, pp. 5385-5397, 2022.

[21] S. A. Mahmoud, I. Ahmad, W. G. Al-Khatib, M. Alshayeb, M. T. Parvez, V. Märgner, and G. A. Fink, "KHATT: An open Arabic offline handwritten text database", *Pattern Recognition*, Vol. 47, No. 3, pp. 1096-1112, 2014.

[22] S. K. Jemni, S. Ammar, and Y. Kessentini, "Domain and writer adaptation of offline Arabic handwriting recognition using deep neural

networks", *Neural Computing and Applications*, Vol. 34, No. 3, pp. 2055-2071, 2022.
[23] M. Lu, Y. Mou, C. L. Chen, and Q. Tang, "An efficient text detection model for street signs", *applied sciences*, Vol. 11, No. 13, pp. 1-16,2021.
[24] Y. Zhang, Y. Chen, C. Huang, and M. Gao, "Object detection network based on feature fusion and attention mechanism", *Future Internet*, Vol. 11, No. 1, pp. 1-14, 2019.
[25] H. Zhao, J. Li, J. Nie, J. Ge, S.Yang, L. Yu, Y. Pu, and K. Wang, "Identification Method for Cone Yarn Based on the Improved Faster R-CNN Model", *Processes*, Vol. 10, No. 4, pp. 1-21, 2022.
[26] X. Cheng, L. Tan, and F. Ming, "Feature Fusion Based on Convolutional Neural Network for Breast Cancer Auxiliary Diagnosis", *Mathematical Problems in Engineering*, Vol. 2021, pp. 1-10, 2021.
[27] X. Renjun, Y. Junliang, W. Yi, and S. Mengcheng, "Fault Detection Method Based on Improved Faster R-CNN: Take ResNet-50 as an Example", *Geofluids*, Vol. 2022, pp. 1-9, 2022.
[28] C. Cao, B. Wang, W. Zhang, X. Zeng, X. Yan, Z. Feng, Y. Liu, and Z. Wu, "An Improved Faster R-CNN for Small Object Detection", *IEEE Access*, Vol. 7, pp. 106838-106846, 2019.
[29] Z. Guo, Y. Tian, and W. Mao, "A Robust Faster R-CNN Model with Feature Enhancement for Rust Detection of Transmission Line Fitting", *Sensors*, Vol. 22, No. 20, pp. 1-16, 2022.
[30] Z. Zheng, P. Wang, D. Ren, W. Liu, R. Ye, Q. Hu, and W. Zuo, "Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation", *IEEE Transactions on. Cybernetics*, Vol. 52, No. 8, pp. 8574-8586, 2021.
[31] C. Huang, A. Yu, and H. He, "Using combined Soft-NMS algorithm Method with Faster R-CNN model for skin lesion detection", In: *Proc. of 6th International Conference on Robotics and Artificial Intelligence*, Singapore, pp. 5-8. 2020.
[32] S. Albahli, M. Nawaz, A. Javed, and A. Irtaza, "An improved faster-RCNN model for handwritten character recognition", *Arabian Journal for Science and Engineering*, Vol. 46, No. 9, pp. 8509-8523, 2021.
[33] M. M. Al-Taee, S. B. H. Neji, and M. Frikha, "Handwriting Arabic Words Recognition in KHATT Dataset Based on Faster R-CNN", In: *Proc. of 2023 6th International Conference on Engineering Technology and its Applications (IICETA)*, Iraq, Al-Najaf, pp. 434-439, 2023.
[34] Tzutalin, "labelImg," 2015. https://github.com/tzutalin/labelImg.