Power System Technology
ISSN:1000-3673

Received: 5-2-2024 *Revised: 15-2-2024* *Accepted: 17-2-2024*

Classification and Localization of Arabic Handwritten Text in KHATT Dataset Based on Faster R-CNN

[1*]May Mowaffaq AL-Taee, [2]Sonia Ben Hassen Neji, [3]Mondher Frikha, [4]Salah Taha Allawi

[1,2,3]École Nationale d'Électronique et de Télécommunications de Sfax, university of Sfax, Tunisia

[4]Department of Computer Science, College of Science, Mustansiriyah University, Baghdad, Iraq

[1]may.tai@enetcom.u-sfax.tn, [2]sonia.benhassen@enetcom.usf.tn,
[3]mondher.frikha@enetcom.usf.tn, [4]salah.taha@uomustansiriyah.edu.iq

Abstract

Arabic handwriting recognition is an important research area in computer vision. Due to the complexity of the Arabic script, this is an arduous task. Several approaches have been proposed to address this challenge, including deep learning algorithms. Despite their efficiency, these approaches present some limitations such as the use of lexicon-driven models, the need of a lot of data for training and the huge computational cost. We propose, in this paper, two novel models based on the robust Faster Region-Convolution Neural Network (Faster R-CNN) to recognize Arabic handwritten sentences. The Faster R-CNN is commonly used to detect objects in images and depends on region proposal algorithms. These models use the pre-trained VGG 16 and ResNet50. Moreover, they use a soft non-maximum suppression strategy (Soft-NMS) in the post-processing stage instead of the traditional NMS to increase the detection rate of overlapping items. The models will be trained independently to identify words in the text lines and tested using sentences from the (KFUPM Handwritten Arabic TexT) KHATT dataset. It will be shown that the faster R-CNN models allow greater accuracy and effectiveness in handwriting recognition. They achieve accuracy rates of 99% and 98% for the two networks, VGG16 and ResNet50, respectively.

Keywords: Faster R-CNN; KHATT dataset; Feature extraction; network Object detection; Arabic handwriting; Soft NMS.

1. Introduction

Handwritten text recognition is a machine's ability to read and understand a handwritten text from various input devices, such as images, paper documents, touchscreens, and other devices [1]. There are two main approaches for recognizing handwritten letters, digits, and words: offline and online techniques. Online systems use sensors to capture data during the writing process, while offline systems rely on images of the user's handwriting taken from a scanner or digital camera. Research suggests that online recognition has a higher recognition rate than offline mode [2],[3]. Many methods have been proposed for offline Arabic handwriting

Power System Technology
ISSN:1000-3673
Received: 5-2-2024 Revised: 15-2-2024 Accepted: 17-2-2024

recognition to convert Arabic writings into a machine-readable format. Arabic handwriting poses more significant challenges for recognition compared to Latin, Japanese, and Chinese because of many factors such as overlaps, touching words, text-line inclination, ligatures, uneven spaces between words, words without dots, and other elements [4]–[6].

Recent advancements in deep learning technology have significantly Influenced the field of Arabic text recognition, leading to many solutions based on deep learning approaches [7]. The progress made in deep learning (DL) has resulted in remarkable advancements, particularly in computer vision. Convolutional Neural Networks (CNNs) are one of the most popular and widely used DL techniques are specialized neural networks that use convolution instead of standard matrix multiplication in at least one of their layers. CNN deals with classification, recognition, multi-object detection, item localization, and handwriting recognition, and some popular CNN architectures include ResNet, VGG, AlexNet, and GoogleNet. Recurrent neural networks (RNNs) like Long Short-Term Memory (LSTM) networks, restricted Boltzmann machines (RBM), Deep Belief Networks (DBN), and Hidden Markov Model (HMM) are among the various deep learning methods extensively employed in handwriting recognition tasks [3], [7]–[11]. Despite their efficiency, these methods suffer from many drawbacks. In fact, they require large training samples, which is computationally expensive. Moreover, some models need pre-segmented text line images and paragraph images in addition to the words to learn. Furthermore, most models use lexicon-driven models including an underlying dictionary that should contains the words to be recognized to ensure good recognition accuracy [12]. Finally, these models frequently suffer from overfitting and thus various regularization approaches must be applied to avoid it [13].

In object detection, CNNs have achieved state-of-the-art advancements in frameworks. Object detection usually consists of two steps: searching for an object in the image and locating it using the bounding box [11]. Object detection algorithms built on deep learning methods are classified into two categories based on the number of stages used to achieve the task: single-stage object detection approaches and two-stage object detection approaches. You Only Look Once (Yolo) and single shot detection model (SSD) are the most popular one-stage object detection algorithms used today. Although the one-stage methods are very fast for real-time usage, they are less efficient than the two-stage ones that have usually greater recognition accuracy [14]. Two-stage object detection techniques include R-FCN, Fast R-CNN, and Faster R-CNN. The progression from R-CNN [15] to Fast R-CNN [16] and then to Faster R-CNN [17] validates the advancements made in this field. Faster R-CNN is a popular model for object recognition and detection. It surpasses previous convolutional neural network (CNN) based designs regarding object detection accuracy [18], [19].

This study proposes a Faster RCNN algorithm that can deliver detection and classification accuracy on par with the latest techniques. Below is a summary of this paper's main contributions:

1. This work considers the first attempt to use Faster R-CNN with the KHATT dataset.
2. Proposing two novel models based on the Faster R-CNN that use the pre-trained VGG 16 and ResNet50.
3. Using Soft-NMS instead of NMS in the final stage of the Faster R-CNN to enhance its object detection efficacy.

Power System Technology
ISSN:1000-3673
Received: 5-2-2024 Revised: 15-2-2024 Accepted: 17-2-2024

4. Using multiple threshold values and comparing their respective results to determine the best value for the IoU metric that increases the detection of words.

The paper is structured as follows: Section 2 will review the main previous techniques developed for recognizing Arabic language in the KHATT dataset. Section 3 will describe the suggested method's components. Section 4 will examine the experimental results. Finally, in section 5, conclusions and some perspectives will be presented.

2. Literature Survey

This section focuses on the recognition systems developed for the Arabic language in the KHATT dataset.

Riaz et al. [20] presented in 2017 a method for Arabic character recognition based on Multi-Dimensional Long Short-Term Memory (MDLSTM), with the Connectionist Temporal Classification (CTC) layer as the last layer. They employed text-line preprocessing to remove excess white space, correct skew, eliminate unnecessary information, and normalize the text. On text lines from the KHATT dataset, they reported a Character Error Rate (CER) of 24.25%, and the system achieved a total accuracy rate of 75.8% on unique text lines in the KHATT dataset. Furthermore, they proposed, in [21] in 2020 improved their previous method by applying five data augmentation techniques and a deep learning strategy. They achieved a higher accuracy rate of 80.02% for character recognition, as well as a character error rate of 4.22% and a label error rate of 19.98%.

Moreover, Liangke et al. [22] developed a novel approach for handwriting recognition that uses reinforcement learning. A CNN was employed to extract features, while a sequence-to-sequence strategy transcribed handwritten text lines. The system incorporated a Policy Network (PN) for reinforcement learning, trained to dynamically select an optimal context length from a range of available context lengths. The encoders and decoders were implemented using Long Short-Term Memory (LSTM). Preprocessing the input images involved converting them to binary images and adjusting them to a 32 pixels size to preserve the aspect ratio. The system can recognize lines of handwritten text from three different datasets, namely IAM, RIMES, and KHATT. In line recognition tasks on the KHATT dataset, the model achieved a 6.93% CER. Overall, the results show that the proposed method is a promising novel approach to handwriting recognition.

Mohamed et al. [23] presented a fully convolutional network (FCN) architecture for unconstrained text recognition. It used depth-wise convolutions, gate blocks, and general data augmentation approaches, trained on full line or word labels with the CTC loss function. They achieved the best results on seven public benchmark datasets, including handwriting, CAPTCHA, OCR, license plate recognition, scene text recognition, and the ICFHR2018 Competition on Automated Text Recognition on a READ dataset. On the KHATT, it achieved a rate of 8.7% on CER and a higher accuracy rate of 91.02%.

Sana et al. [24] in the same year proposed, a method for recognizing handwritten Arabic text using various combination architectures involving Bidirectional Long Short-Term Memory (BLSTM) and CTC. The researchers compared three levels of combinations: low-level fusion, mid-level combination methods, and high-level fusion trained in various handcrafted features. Their study employed the Arabic KHATT dataset for experimentation

Power System Technology
ISSN:1000-3673
Received: 5-2-2024 Revised: 15-2-2024 Accepted: 17-2-2024

and applied the preprocessing step, which includes the line image conversion into a binary image, height normalization, denoising, and deflection angle determination. The results showed that cooperative systems, with high-level combinations, outperformed individual methods. The test set achieved a Word Error Rate (WER) of 13.52% and a CER of 7.85%, corresponding to an accuracy of 86.48%. After, in 2019, Sana et al. [25] proposed a novel approach for recognizing any out-of-vocabulary (OOV) word as an arbitrary sequence of sub-word units. This approach was tested on two handwriting recognition methods: one based on customized HOG features and a BLSTM and CTC architecture, and the other based on CNN's learned features and the MDLSTM as a classifier with CTC. The experiments were conducted on the KHATT Arabic dataset after applying the preprocessing step, which includes converting the line image into a binary image, height normalization, denoising, and deflection angle determination. Authors showed that combining full-word models of character, morpheme, and PAWs is successful for dealing with OOV words, particularly when employing the CNN-MDLSTM architecture. Sub-word and character language models (LMs) can assure significant coverage of OOVs. The Combination of Full-Word, Morphemes, Paws, and Character model achieved a WER of 20.86%. Furthermore, they proposed, in [26], an Arabic handwritten text recognition model using KHATT and AHTID/MW databases. As part of the preprocessing step, the input gray image was standardized to a fixed height of 96. The model comprises convolutional and MDLSTM layers and a combination of CTC and WFST to convert the output into a sequence of words, morphemes, or parts of Arabic words (PAWs). The authors used three different recognition methods to recognize (OOV) words and recovered them using a dynamic lexicon. Overall, they achieved a KHATT word error rate (WER) of 20.83% and an accuracy of 79.17%.

In addition, Zouhaira et al. [27] proposed, in 2019, a method for recognizing Arabic handwritten text lines using a combination of two deep learning algorithms, CNN and BLSTM. They divided the system into three phases. The first phase is preprocessing, which includes removing white areas, binarization, skew detection, and correction on the text line. The second phase is feature extraction using CNN and BLSTM sequence modelling. Finally, they used the CTC function for text recognition. Evaluated on the KHATT dataset, the system achieved an 8.63% CER and a 20.17% WER, resulting in an accuracy of about 79.83%. Recently, in 2020, Zouhaira et al. [28] proposed an effective open-vocabulary offline recognition method for handwritten Arabic text based on a character model. They firstly improved image quality through various preprocessing steps, including binarization, resizing, baseline detection, line skew correction, slant correction, and normalization of character height. Then, they developed a deep RCNN model that uses a VGGNet architecture combined with a BLSTM layer to learn an open vocabulary. A CTC beam search decoder with BLSTM was used for sequence modelling. The experiments were conducted using KHATT with AHTID/MW databases. The proposed model achieved an accuracy of 87.39%, a CER of 2.43%, and a WER of 12.61%. Moreover, they presented, in [29], a recognition model for Arabic script using the transfer learning (TL) approach. The CNN-BLSTM-CTC architecture underpins the recognition system. Three Arabic text line databases were used in the paper which are P-KHATT, KHATT and AHTID. Transfer learning from the printed P-KHATT database was chosen for training to the handwritten KHATT and AHTID databases. The experiments revealed that the TL technique performed well. For KHATT dataset, the system achieved a CER of about 1.64%

Power System Technology
ISSN:1000-3673
Received: 5-2-2024 Revised: 15-2-2024 Accepted: 17-2-2024

and a WER of 10.22%. More recently, [30] in 2021 they proposed a novel character recognition system that improves previous recognition models. The system was divided into three phases: preprocessing, feature extraction, and recognition. Preprocessing involves eliminating any variable resources from the image scanning phase and normalizing the size of the images. CNN architectures were used to extract relevant image descriptors. Recognition included sequence modelling and training using the BLSTM network and the CTC decoder. The proposed system was trained and evaluated using two Arabic text recognition datasets, KHATT, and AHTID/MW. It achieved, on the KHATT dataset, a WER of about 11.53% with Dense-VGGNet and a recognition rate of 89%. The results show that the CRNN based on densely VGGNet-dense and BLSTM is a very interesting model for Arabic handwritten recognition.

Takwa et al. [31] proposed in 2022 an attention-based CNN-Att-BLSTM-CTC architecture for extracting handwritten Arabic words. First, the significant features of the text-line input image were retrieved using a CNN. The extracted features and the text line transcription were then sent into an attention-based BLSTM network to propagate information. The BLSTM made features sequence in order while the attention mechanism selected relevant information from the features sequences. Finally, a CTC was used to learn the alignment between text-line images and their transcription automatically. Authors trained the suggested model on KHATT. The experimental results showed a rate efficiency of 91.7%.

Despite the efficiency of the existing models, they suffer from some limitations. In fact, these models use several preprocessing steps like binarization, skew detection and correction, etc. A sliding window is also employed in the features extraction step. Moreover, many methods use lexicons, and language model for the recognition**.** Furthermore, these methods need a lot of training data to ensure good recognition accuracy.

3. Methodology of Faster R-CNN for Handwriting Word Detection

Our proposed method is a deep learning model based on the Faster Region-Convolution Neural Network (Faster R-CNN) [17]. It is designed to detect and classify Arabic words in a text line from the KHATT dataset. The method is divided into two stages. In the first stage, we will prepare the data by removing the white areas, normalizing the size of the images, and then labeling the regions of interest in each image to determine the words on which we will train the model. The second stage includes applying the Faster R-CNN to detect and classify the words in each image. Figure 1 shows the general framework of the proposed model.

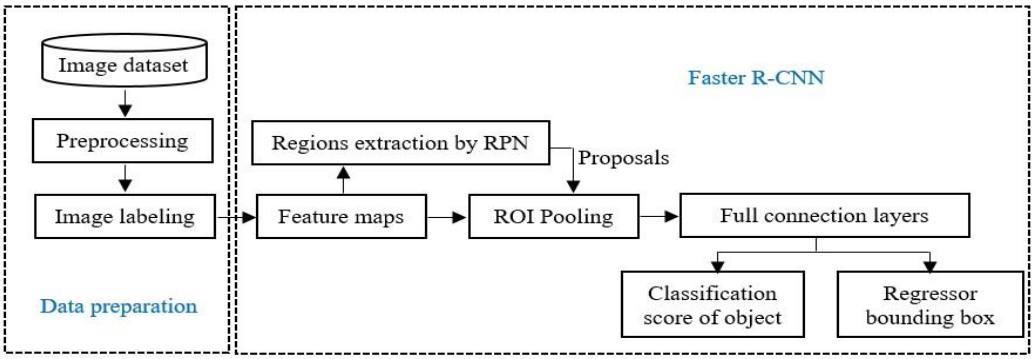


Figure 1. General framework of the proposed model

Power System Technology
ISSN:1000-3673
Received: 5-2-2024 Revised: 15-2-2024 Accepted: 17-2-2024

3.1. Preparation of Data

3.1.1. Dataset

KHATT (KFUPM Handwritten Arabic TexT) database was presented at the 13th International Conference on Frontiers in Handwriting Recognition (ICFHR) in 2012, intending to facilitate the researches in character recognition regarding Arabic script [32], [33]. It is a freely available offline handwritten text comprising 4000 paragraphs written by 1000 writers from different countries, age groups, genders, and education levels. This dataset contains unrestricted writing styles [20], [34]. It includes 2000 unique, randomly selected paragraphs with different text contents and 2000 fixed paragraphs with the same content [35]. The paragraphs are segmented into 9000 text lines automatically [32]. The database is very suitable for research in Arabic writer identification and handwriting recognition [36]. Figure 2, given below, shows some samples of the text lines in the KHATT dataset. Researchers face various difficulties when dealing with this database. The main problems are the following:

1. Many text-line images have different extra white regions;
2. There is no proper baseline;
3. Many of the text lines are skewed;
4. Each text line has a different height;
5. There is no set number of words per text line.

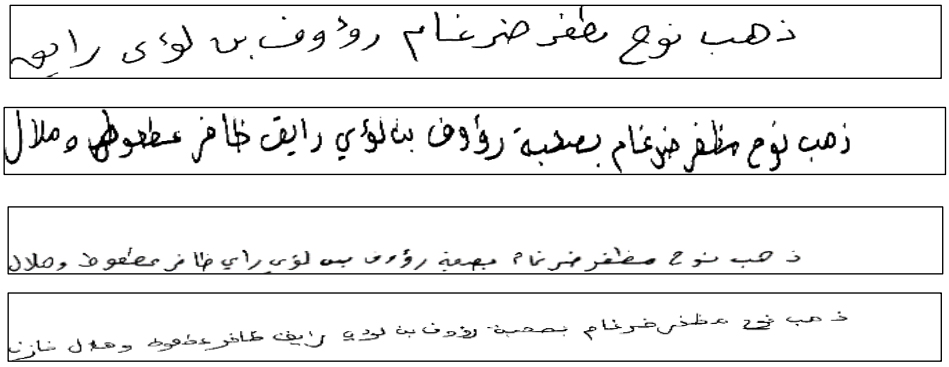
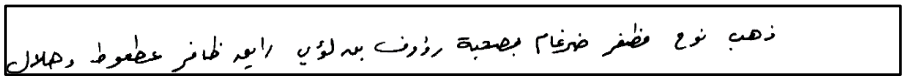


Figure 2. Samples of the text lines in the KHATT dataset.

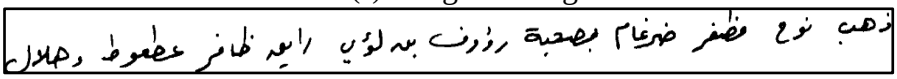
3.1.2. Preprocessing

In the first step, the images are processed to remove all white areas, and the maximum width and height values are determined. Then, in the second step, a white box, of dimensions equal to the maximum width and height values with 5 pixels added on each side, is generated. This facilitates the labeling process around each category in the image. In the third step, the text image is positioned at the center of the white box, resulting in a final composite image with a size of 1424×248 pixels used during the training and testing phases of the model. Figure 3 shows the steps of removing the white areas from the original image and inserting the resulting text image into the white image. Moreover, algorithm 1 given below describes the general steps for the image preprocessing.

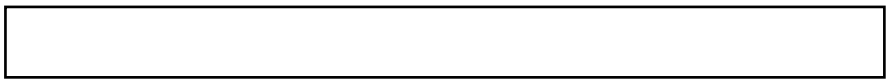
Power System Technology
ISSN:1000-3673
Received: 5-2-2024 Revised: 15-2-2024 Accepted: 17-2-2024



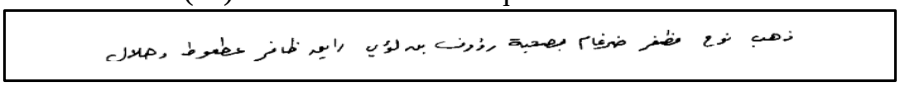
(a) Original image



(b) Image after removing white space



(c) White box with adequate dimensions



(d) New image after putting the text in the white box

Figure 3. Preprocessing steps

Algorithm 1: Image Preprocessing

Input: Original images with non-uniform size

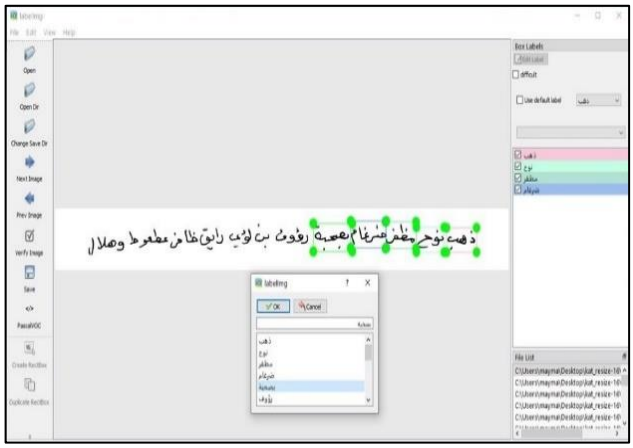
Output: Images with uniform size

1. Input images with non-uniform size.
2. Identify and remove white areas from images.
3. Calculate the width and height values for all images.
4. Find the max value of the width and height.
5. Calculate the new size by adding 5 pixels to the max width and height value to facilitate the labeling process around each category in the image.
6. Create a white image by using the new size.
7. Create new images by adding the images after deleting the white areas in the middle of the white image.

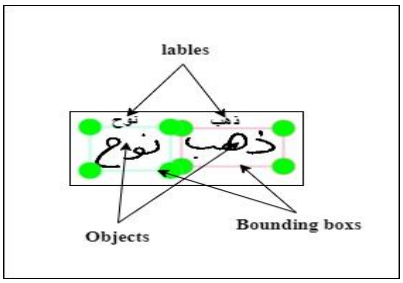
3.1.3. Data Labeling

To train the proposed model, we require two types of input: the image sample and the word's location in the input image. The KHATT dataset used in our work does not contain the required annotations. To address this issue, the labelling tool is used manually to construct bounding box (bbox) annotations around each object in the image [37]. The resulting annotations are saved in XML files, which contain the bbox values and class names of each object (xmax, xmin, ymax, ymin, height, and width). Each image has its own XML file. Finally, the XML files are grouped into one CSV file, and then converted to a TXT file used for the training and testing phases. Figure 4 shows the steps of labeling the objects in the image.

Power System Technology
ISSN:1000-3673
Received: 5-2-2024 Revised: 15-2-2024 Accepted: 17-2-2024



a) LabelImg tool



b) Labeling classes in the image

	filename	width	height	class	xmin	ymin	xmax	ymax
2	Khat1.tif	2424	248	ذهب	2183	92	2315	158
3	Khat1.tif	2424	248	نوح	2095	97	2188	172
4	Khat1.tif	2424	248	بن	1398	106	1468	176
5	Khat1.tif	2424	248	رؤوف	1466	106	1601	186
6	Khat1.tif	2424	248	رايق	1183	97	1291	172
7	Khat1.tif	2424	248	عطعوط	873	104	1056	172
8	Khat1.tif	2424	248	هلال	725	94	844	165
9	Khat1.tif	2424	248	بصحية	1600	106	1786	172
10	Khat1.tif	2424	248	مظفر	1946	84	2088	174
11	Khat1.tif	2424	248	ضرغام	1781	91	1941	166
12	Khat1.tif	2424	248	لؤي	1291	96	1396	169
13	Khat1.tif	2424	248	ظافر	1061	96	1188	176
14	Khat1.tif	2424	248	و	826	126	873	174
15	Khat1.tif	2424	248	خازن	586	86	738	161
16	Khat1.tif	2424	248	عفيف	428	91	568	164
17	Khat1.tif	2424	248	للحج	301	96	438	179

c) CSV file

File Edit Format View Help
khat1.tif,2133,81,2261,159,ذهب
khat1.tif,2034,89,2134,172,نوح
khat1.tif,1408,91,1551,177,رؤوف
khat1.tif,1331,92,1412,175,بن
khat1.tif,1114,89,1232,187,رايق
khat1.tif,812,77,995,167,عطعوط
khat1.tif,593,75,744,150,هلال
khat1.tif,1561,99,1722,175,بصحية
khat1.tif,1893,71,2023,171,مظفر
khat1.tif,1726,84,1896,176,ضرغام
khat1.tif,1231,91,1333,181,لؤي
khat1.tif,993,81,1126,176,ظافر
khat1.tif,743,106,791,169,و
khat1.tif,451,69,601,154,خازن
khat1.tif,323,69,458,151,عفيف
khat1.tif,191,66,326,179,للحج

d) TXT file

Figure 4. Steps of labeling classes in images

3.2. Faster R-CNN

In our work, we consider two pre-trained CNN models, VGG16 and ResNet50, which are the most frequently used with Faster R-CNN. These modules are trained independently to extract features from the labeled regions, allowing us to identify each word's unique characteristics. Then, the Region Proposal Networks (RPN) is applied to generate exact regional proposals. Finally, the Fast R-CNN method (detector) is used. It involves Region Proposal Network (RoI) Pooling, class localization, and classification to identify the classes with scores for words and their locations (the bounding boxes around words) within the images [38], [39]. Soft-NMS was used instead of Non-Maximum Suppression (NMS) in post-processing to improve the detection accuracy of the word [40]. Algorithm 2 summarizes the general outline of the Faster R-CNN for detecting and classifying Arabic words.

Power System Technology
ISSN:1000-3673
Received: 5-2-2024 Revised: 15-2-2024 Accepted: 17-2-2024

Algorithm 2: Faster R-CNN to detect and classify Arabic handwritten words

Input: Image of an Arabic handwritten line of text

Output: Detected and classified Arabic words in the image with their bounding boxes and confidence scores

1. Preprocess the images.
   a. Removing the white areas
   b. Normalizing the size of the images.
   c. Labeling the classes in all images.

2. Apply the Faster R-CNN
   a. Extracting the feature maps using a convolutional neural network (CNN).
   b. Extract the proposed regions by using the region proposal network (RPN).
   c. Extract the RoI feature maps using the RPN and feature maps.
   d. Extract the class probabilities and bounding box coordinates by passing the RoI feature maps to the fully connected layers.
   e. Calculate the loss between the predicted class probabilities, bounding box coordinates, ground truth class labels, and bounding boxes.
   f. Update the parameters of the network using backpropagation and gradient descent.
   g. Repeat steps (a-f) for multiple epochs until convergence.

3. Use the trained model to predict the class labels and bounding boxes for new handwritten text images.

3.2.1. Features' Extraction using the shared CNN

Our approach uses CNN, a leading-edge object detection technique. CNNs employ a set of convolution and pooling operations to extract essential features from images. In our method, we pass each image and its corresponding annotations through a feature extractor algorithm, resulting in the generation of a feature map [41], [42]. For this purpose, we employ pre-trained networks trained on the ImageNet dataset to ensure efficient and effective extraction of image features. The networks used in our approach are described below.

- **VGG16 Network**

Simonyan [43] introduced the VGG16 network architecture. This network was used to extract image features [44]. It is a pre-trained ConvNet with a 16-depth weight layers comprising 13 convolution layers with Relu (Rectified Liner Units) activation functions and 3 fully connected layers. Moreover, it contains 4 pooling layers [45]. The network core is created by removing the final fully connected layer and only keeping the front part of the convolutional layer [46].

Power System Technology
ISSN:1000-3673
Received: 5-2-2024 Revised: 15-2-2024 Accepted: 17-2-2024

- **Residual Network (ResNet)**

It is a network architecture that was proposed in 2015 [47]. ResNet has been extensively employed in various domains, including recognition, detection, segmentation, classification, and object detection. This network can address the clear deterioration issue as network depth increases [48], [49]. The ResNet50 is a feature extraction algorithm pre-trained on the ImageNet dataset. In the ResNet50-based residual module, we can find two distinct structures: the identity block and the convolution block. The Identity Block module maintains the same dimensions as its input and output vectors, facilitating the direct deepening of the network. Through concatenation, deep semantic information is learned. The ConvBlock module uses a 1×1 convolution process to ensure compatibility between its input and output vectors, adjusting the dimensions accordingly. ResNet50 comprises four sets of residual modules, one fully connected layer, and one convolutional layer [48].

3.2.2. Region Proposal Network (RPN)

The RPN, which stands for Region Proposal Network, is a fully convolutional network (FCN) that can be trained from end to end. Its purpose is to generate exact regional proposals. The detection network and the RPN in this system use shared full-image convolutional features, enabling the generation of roughly cost-free region proposals [50]. These proposals can then be fed into the Fast R-CNN for detection [51]. The RPN takes the feature map output from the previous network as input to generate the proposals, processes them, and outputs object proposals. It uses a 3×3 sliding window approach to process the input and generate a feature vector. At each image point, 9 anchors are generated for each sliding window, with three aspect ratios (1:1, 2:1, 1:2) and three scales (32, 64, and 128), but in the exact centre. Two fully connected layers then process proposals to determine the likelihood that an object will be present in the proposed window. One layer is dedicated to regression and predicts the object's bounding box coordinates. While the other layer determines if the proposal is an object (word) or it is a background [48]. Figure 5 illustrates the Region Proposal Network (RPN). The RPN guides the Fast R-CNN module on where to look.

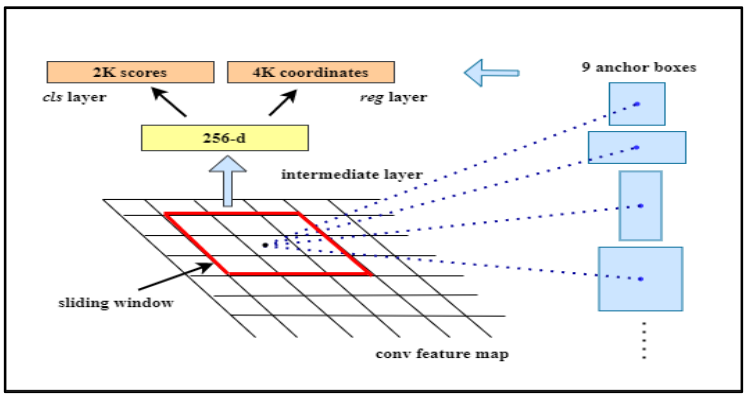


Figure 5. Region Proposal Network (RPN) [17]

Power System Technology
ISSN:1000-3673
Received: 5-2-2024 Revised: 15-2-2024 Accepted: 17-2-2024

Intersection over Union (IoU) is a key object detection indication. In regression, the IoU is the best indication of the predicted bounding box's distance from the truth box. Equation 1 shows the IoU formula [52].

$$IoU = \frac{Anchor \cap GTBox}{Anchor \cup GTBox} \quad (1)$$

Where: IoU is the ratio between the area of overlap of the ground truth bounding box (GTBox) and the anchor in question and their union, Anchors are output suggestions that receive an objectness score based on the intersection over union (IoU) score [53].

The RPN uses two kinds of anchors: positive and negative. An anchor is positive when the IoU score exceeds 0.7 with any ground truth box. A negative anchor is assigned when the IoU score is lower than 0.3 for each ground truth area. The anchors whose scores are between 0.3 and 0.7 do not affect the training loss, while the remaining negative and positive anchors are used to train the next network module [14]. Equation 2 is used to assign positive or negative anchors based on the threshold value [54].

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

The loss function of the whole network is given by the following equation:

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

Where i is the index of anchors, pi is the probability that the i anchor is predicted to be the true label, pi* is the presence or absence of a target for the anchor, ti is the prediction of the bounding box regression parameter of the i anchor, ti* is the ground truth box corresponding to the i anchor, Ncls is the batch size, Nreg is the number of anchor positions, and λ is the balance parameter. Lcls is a binary log loss and Lreg is a smoothed L1 loss.

Faster R-CNN can be trained end-to-end by back-propagation using the stochastic gradient descent (SGD) for the optimization of the loss function [14], [17], [51], [55]. NMS is a critical step in object detection models that aims to reduce redundancy in proposals generated by RPNs. NMS selects the detection box with the highest classification score based on CLS scores and eliminates other boxes with significant overlap surpassing a predefined IoU threshold. By discarding redundant proposals, NMS reduces the number of proposals while maintaining detection accuracy [17], [53], [56]. To address the multiple detections of the same object in an image, we used Soft-NMS after the classification stage, which offers many benefits. In fact, it improves handling of overlapping detections and enhances localization accuracy in addition to the flexibility in the selection of bounding boxes, and the ability to fine-tune confidence scores. These advantages make soft-NMS a valuable technique for refining object detection in object detection models like Faster R-CNN [40].

Power System Technology
ISSN:1000-3673
Received: 5-2-2024 Revised: 15-2-2024 Accepted: 17-2-2024

3.2.3. Fast R-CNN detector

A detection network receives the feature map and the regions of interest generated by the previous networks as input. It comprises a classification layer and a bounding box regression layer. The classification layer predicts the class probabilities for each region of interest, while the bounding box regression layer predicts the coordinates of the bounding box for each region of interest. Together, these layers generate the predicted class and bounding box coordinates for the detected objects [55], [57].

- Region of interest pooling (RoI pooling)

For every RoI from the input, RoI pooling takes a section of the corresponding feature map and scales it to fixed size. Then, after processing the feature maps and proposals, summing these information, and generating proposal feature maps with fixed sizes, the latter will be reshaped into vectors to be fed to fully connected layers [38], [57].

- Classification and regression

The classifier unit, which is a fully connected layer, displays the class associated with each word while the bbox formed via bounding box regression reveals the eventual position of the recognized word [38], [41], [51].

4. Implementation and Results

In this stage, we use a dataset of 1000 images with a size of 2424×248 pixels, each with varying counts of classes, all in ".tif" format. Split into 800 images for training and 200 images for testing. We trained our proposed method to localize and classify 16 specific classes. The number of epochs is 35 for VGG16 and ResNet50, each comprising 800 iterations, with a learning rate 1e-5.

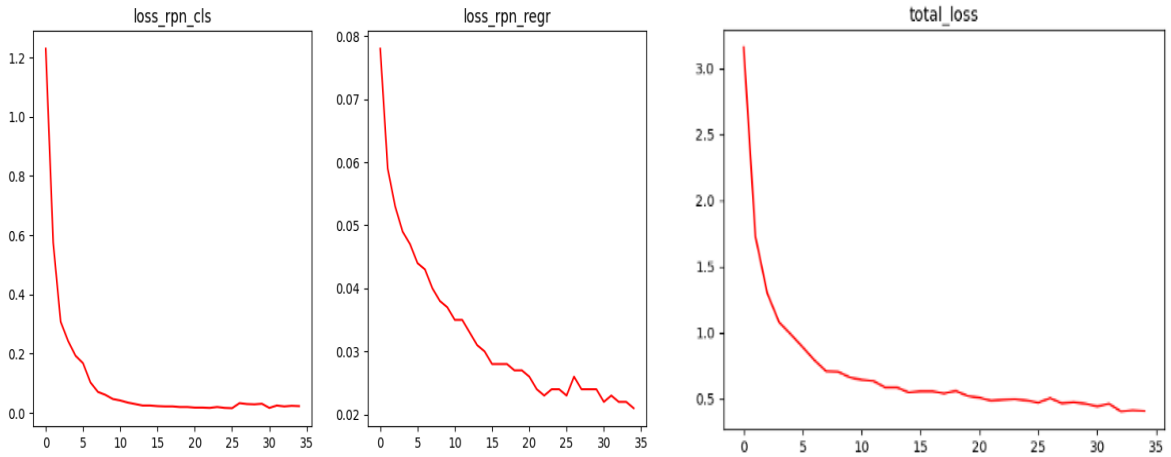
The training process executes on an NVIDIA Processor Core i9 and uses Python to write the codes. With the VGG16 model, the Faster R-CNN requires 76 hours to be trained while with the ResNet50 model, it requires only 41 hours the performance. The entire training is carried out in a CPU environment. We note here that the training time can be significantly reduced if we use a GPU environment.

To evaluate the performances of these models in terms of detection and classification. The experiments are conducted on a testing images containing (2763) words. Each image varies in terms of number of words, percentage of distortions, and writing style. Table 1 shows the names of each class used in the train and test phase and the total number of each class. Figures 6 and 7 illustrate the total loss for the two models, VGG16 and ResNet50.

Power System Technology
ISSN:1000-3673
Received: 5-2-2024 Revised: 15-2-2024 Accepted: 17-2-2024

Table 1. Classes' names and counts for the training and testing stage

Training stage		Testing stage	
Class name	**Class count**	**Class name**	**Class count**
ذهب	800	ذهب	200
نوح	798	نوح	200
مظفر	788	مظفر	197
ضرغام	795	ضرغام	199
بصحبة	786	بصحبة	197
رؤوف	796	رؤوف	200
بن	797	بن	195
لؤي	795	لؤي	199
رايق	778	رايق	194
ظافر	766	ظافر	195
عطعوط	733	عطعوط	190
و	598	و	173
هلال	591	هلال	168
خازن	410	خازن	122
عفيف	263	عفيف	80
للحج	161	للحج	54

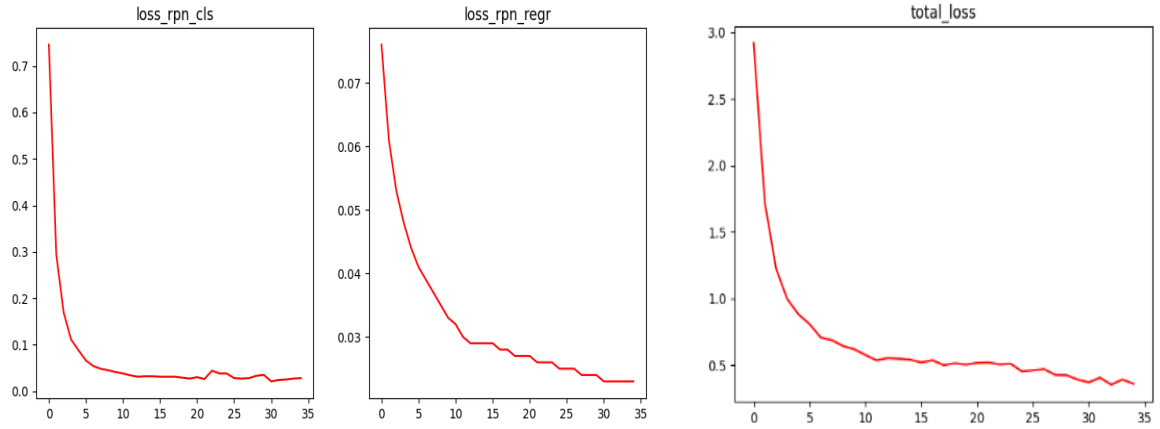


a) RPN loss b) Total loss

Figure 6. Training loss in the VGG16 model

Power System Technology
ISSN:1000-3673

Received: 5-2-2024 *Revised: 15-2-2024* *Accepted: 17-2-2024*



a) RPN loss b) Total loss

Fig. 7. Training loss in the ResNet50 model

Figure 8 shows some results obtained from the test phase using the VGG16 network after training the model with 35 epochs and 800 iterations.

Figure 8. Some test results for the Faster R-CNN with the VGG16 network.

Figure 9 shows some results obtained from the test phase using the ResNet50 network after training the model with 35 epochs and 800 iterations.

Figure 9. Some test results for the Faster R-CNN with the Resnet50.

Power System Technology
ISSN:1000-3673
Received: 5-2-2024 Revised: 15-2-2024 Accepted: 17-2-2024

We generate the confusion matrices for the two proposed models. Then, we use several evaluation metrics to assess the proposed approach's efficiency, including accuracy, precision (P), recall (R) and F1_score (F). These values depend on the IoU threshold which is a crucial parameter that determines whether a predicted bounding box is a true positive or a false positive.

We use several evaluation metrics to assess the proposed approach's efficiency, including accuracy, precision, recall, F1 score These metrics are defined as follows:

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

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

$$F1\_Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (7)$$

Determining the appropriate IoU threshold value depends on the type and purpose of the application. In our study, we choose three IoU threshold values (0.4, 0.45, and 0.5) and compare the results to determine the best value that suits our work. There values are the frequently used in the literature [58], [59]. In fact, when the IoU threshold is adjusted from 0.5 to slightly lower values, such as 0.45 and 0.40, the detector tends to predict more small objects included in the training dataset, and thus the detection performance will be improved. Tables 2 and 3 present the results of applying three different threshold IoU values for the VGG16 and ResNet50, respectively.

We also use, for the evaluation of the proposed models, the mean average precision (mAP) which is an essential metric in target detection employed to assess the model's quality. It is defined as:

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \quad (8)$$

Where (N) is the total number of classes, and (AP_i) is the average precision of the (ith) class.

Power System Technology
ISSN:1000-3673

Received: 5-2-2024 *Revised: 15-2-2024* *Accepted: 17-2-2024*

Table 2. Results of applying three different threshold IoU values with the VGG16

Class name	Th. IoU values =0.4			Th. IoU values =0.45			Th. IoU values =0.5		
	Precision	Recall	F1_score	Precision	Recall	F1_score	Precision	Recall	F1_score
ذهب	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
نوح	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
مظفر	1.0	0.99	0.99	1.0	0.99	0.99	1.0	0.99	0.99
ضرغام	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
بصحبة	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
رؤوف	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
بن	1.0	0.99	1.0	1.0	0.99	0.99	1.0	0.98	0.99
لؤي	1.0	0.99	1.0	1.0	0.99	1.0	1.0	0.99	1.0
رايق	1.0	0.98	0.99	1.0	0.98	0.99	1.0	0.97	0.99
ظافر	1.0	0.98	0.99	1.0	0.98	0.99	1.0	0.98	0.99
عطعوط	1.0	0.99	1.0	1.0	0.99	1.0	1.0	0.99	1.0
و	1.0	0.98	0.99	1.0	0.94	0.97	1.0	0.85	0.92
هلال	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.99	1.0
خازن	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
عفيف	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
للحج	1.0	0.98	0.99	1.0	0.98	0.99	1.0	0.98	0.99
Accuracy	99.4			99.1			98.4		

Table 3. Results of applying three different threshold IoU values with the ResNet50

Class name	Th. IoU values =0.4			Th. IoU values =0.45			Th. IoU values =0.5		
	Precision	Recall	F1_score	Precision	Recall	F1_score	Precision	Recall	F1_score
ذهب	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
نوح	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
مظفر	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
ضرغام	1.0	0.99	0.99	1.0	0.99	0.99	1.0	0.99	0.99
بصحبة	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.99	0.99
رؤوف	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
بن	1.0	0.99	1.0	1.0	0.99	1.0	1.0	0.98	0.99
لؤي	1.0	0.98	0.99	1.0	0.98	0.99	1.0	0.98	0.99
رايق	1.0	0.99	0.99	1.0	0.99	0.99	1.0	0.99	0.99
ظافر	1.0	0.99	0.99	1.0	0.99	0.99	1.0	0.99	0.99
عطعوط	1.0	0.98	0.99	1.0	0.98	0.99	1.0	0.98	0.99
و	1.0	0.84	0.91	1.0	0.83	0.91	1.0	0.76	0.86
هلال	1.0	0.98	0.99	1.0	0.98	0.99	1.0	0.98	0.99
خازن	1.0	0.98	0.99	1.0	0.98	0.99	1.0	0.97	0.98
عفيف	1.0	0.98	0.99	1.0	0.98	0.99	1.0	0.98	0.99
للحج	1.0	0.94	0.97	1.0	0.94	0.97	1.0	0.94	0.97
Accuracy	98			98			97.3		

It is shown from the conducted experiments that the best performances for the two models are obtained after 35 epochs. The VGG16 model achieved an accuracy of 99.4% and mAP of 0.999. While the model with ResNet50 achieved an accuracy of 98% and mAP 0f 0.998. Moreover, the best IoU threshold value is 0.4.

Power System Technology
ISSN:1000-3673
Received: 5-2-2024 Revised: 15-2-2024 Accepted: 17-2-2024

This study investigates the utilization of Faster R-CNN with the KHATT dataset to detect specific classes in text lines. Notably, it is essential to acknowledge that the comparison presented here may be biased because of several factors. These factors include adopting different detection and classification models, variations in preprocessing methodologies, and discrepancies in the size of the used data. The comparison primarily emphasizes the accuracy achieved by applying these models to the KHATT dataset. Table 4 provides a comparative analysis of the proposed models' performances with those of other techniques using the KHATT dataset. It can be clearly shown that our two proposed models exhibit better accuracy compared to the other models. In addition to these interesting results, they have some other advantages. In fact, we have not integrated organizational techniques, dictionaries, or linguistic models contrarily to some state-of-the-art models. Moreover, our models do not require several preprocessing techniques and large training samples. Finally, they are not very computationally expensive. Despite these advantages, our methods require manual labeling or annotation of the training images which is a relatively hard task.

Tables 4. Shows the methods used in KHATT database.

Ref.	Model	Accuracy
Ref. [20]	MDLSTM+CTC	75.8%
Ref. [21]	MDLSTM+CTC	80.02%
Ref. [22]	CNN+BLSTM+attention+CTC+reinforcement	93.07%
Ref. [23]	FCN+CTC	91.02%
Ref. [24]	BLSTM+CTC	86.48%
Ref. [25]	CNN+MDLSTM+CTC	79.14%
Ref. [26]	CNN+MDLSTM+CTC	79.17%
Ref. [27]	CNN+BLSTM+CTC	79.83%
Ref. [28]	CNN+BLSTM+CTC	87.39%
Ref. [29]	CNN+BLSTM+CTC	89.78%
Ref. [30]	CNN(dense-VGGNe)+LSTM+CTC	89%
Ref. [31]	CNN+Attention+ConvLSTM+CTC	91.7%
Our models	Faster R-CNN with VGG16	99.4%
	Faster R-CNN with ResNet50	98%

5. Conclusion

This research improves the previous recognition system, which utilizes deep learning techniques and focuses on localizing and recognizing Arabic handwritten words in the KHAAT dataset. The system used the Faster R-CNN architecture and comprises two models using the pre-trained VGG 16 and ResNet50, respectively. The two models are trained independently to extract feature maps from the input images. The work improvement includes increasing the training classes in the text line to 16 instead of 8, using the Soft-NMS instead of the NMS after the classification step stage and decreased the IoU value to choose the best value. Selecting the IoU threshold value that suits the specific application carefully is crucial. Therefore, reducing the IoU threshold results in more true positives and fewer false positives. Soft-NMS offers benefits, such as improved handling of overlapping detections, enhanced localization accuracy, flexibility in the selection of bounding boxes, and the ability to fine-tune confidence scores.

Power System Technology
ISSN:1000-3673
Received: 5-2-2024 Revised: 15-2-2024 Accepted: 17-2-2024

The models achieve 99% and 98% accuracy for VGG16 and ResNet50, respectively, when using the IoU threshold equal to 0.4. In future work, the research will broaden to another dataset, and alternative pre-trained networks will replace the models employed in the currently proposed method, as well as recognize paragraphs.

Acknowledgment
The authors appreciate the help from the National School of Electronics and Communications in Sfax.

Data availability: this work uses the KHATT dataset. It is publicly available, see reference [32, 33].

References

[1] A. Abdallah, M. Hamada, and D. Nurseitov, "Attention-based fully gated cnn-bgru for russian handwritten text," *J. Imaging*, vol. 6, no. 12, 2020, doi: 10.3390/jimaging6120141.
[2] H. Lamtougui, H. El Moubtahij, H. Fouadi, A. Yahyaouy, and K. Satori, "Offline Arabic Handwriting Recognition Using Deep Learning: Comparative Study," *2020 Int. Conf. Intell. Syst. Comput. Vision, ISCV 2020*, 2020, doi: 10.1109/ISCV49265.2020.9204214.
[3] M. M. Al-Taee, S. B. H. Neji, and M. Frikha, "Handwritten Recognition: A survey," *4th Int. Conf. Image Process. Appl. Syst. IPAS 2020*, pp. 199–205, 2020, doi: 10.1109/IPAS50080.2020.9334936.
[4] A. Zafar and A. Iqbal, "Machine Reading of Arabic Manuscripts using KNN and SVM Classifiers," in *2020 7th International Conference on Computing for Sustainable Global Development (INDIACom)*, Mar. 2020, pp. 83–87. doi: 10.23919/INDIACom49435.2020.9083696.
[5] N. AbdAllah and S. Viriri, "Off-Line Arabic Handwritten Words Segmentation using Morphological Operators," *Signal Image Process. An Int. J.*, vol. 11, no. 6, pp. 21–36, 2020, doi: 10.5121/sipij.2020.11602.
[6] M. Awni, M. I. Khalil, and H. M. Abbas, "Deep-learning ensemble for offline arabic handwritten words recognition," *Proc. - ICCES 2019 2019 14th Int. Conf. Comput. Eng. Syst.*, no. May 2020, pp. 40–45, 2019, doi: 10.1109/ICCES48960.2019.9068184.
[7] T. B. A. Gader and A. K. Echi, "Attention-Based Deep Learning Model for Arabic Handwritten Text Recognition," *Mach. Graph. Vis.*, vol. 31, no. 1–4, pp. 49–73, 2022, doi: 10.22630/MGV.2022.31.1.3.
[8] A. AL-Saffar, S. Awang, W. AL-Saiagh, S. Tiun, and A. S. Al-khaleefa, "Deep Learning Algorithms for Arabic Handwriting Recognition: A Review," *Int. J. Eng. Technol.*, vol. 7, no. 3.20, p. 344, 2018, doi: 10.14419/ijet.v7i3.20.19271.
[9] M. N. Aljarrah, M. M. Zyout, and R. Duwairi, "Arabic Handwritten Characters Recognition Using Convolutional Neural Network," *2021 12th Int. Conf. Inf. Commun. Syst. ICICS 2021*, pp. 182–188, 2021, doi: 10.1109/ICICS52457.2021.9464596.

Power System Technology
ISSN:1000-3673
Received: 5-2-2024 Revised: 15-2-2024 Accepted: 17-2-2024

[10] L. Alzubaidi *et al.*, *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*, vol. 8, no. 1. Springer International Publishing, 2021. doi: 10.1186/s40537-021-00444-8.
[11] V. Romanuke, "a Dropout Technique Study for the Faster R-Cnn Detectors With Pretrained Convolutional Neural Networks for Detecting Very Simple Objects That Can Be Masked," *Appl. Math. Informatics*, vol. 26, no. 26, pp. 90–104, 2018, doi: 10.30970/vam.2018.26.9837.
[12] R. Mondal, S. Malakar, E. H. Barney Smith, and R. Sarkar, "Handwritten English word recognition using a deep learning based object detection architecture," *Multimed. Tools Appl.*, vol. 81, no. 1, pp. 975–1000, 2022, doi: 10.1007/s11042-021-11425-7.
[13] N. Alrobah and S. Albahli, "Arabic Handwritten Recognition Using Deep Learning: A Survey," *Arab. J. Sci. Eng.*, vol. 47, no. 8, pp. 9943–9963, 2022, doi: 10.1007/s13369-021-06363-3.
[14] Z. Guo, Y. Tian, and W. Mao, "A Robust Faster R-CNN Model with Feature Enhancement for Rust Detection of Transmission Line Fitting," *Sensors (Basel).*, vol. 22, no. 20, 2022, doi: 10.3390/s22207961.
[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2014, pp. 580–587. doi: 10.1109/CVPR.2014.81.
[16] R. Girshick, "Fast R-CNN," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, pp. 1440–1448, Apr. 2015, doi: 10.1109/ICCV.2015.169.
[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.
[18] A. Jindal and R. Ghosh, "Text line segmentation in indian ancient handwritten documents using faster R-CNN," *Multimed. Tools Appl.*, vol. 82, no. 7, pp. 10703–10722, 2023, doi: 10.1007/s11042-022-13709-y.
[19] M. M. Al-Taee, S. Ben Hassen Neji, and M. Frikha, "Handwriting Arabic Words Recognition in KHATT Dataset Based on Faster R-CNN," in *2023 6th International Conference on Engineering Technology and its Applications (IICETA)*, Jul. 2023, pp. 434–439. doi: 10.1109/iiceta57613.2023.10351215.
[20] R. Ahmad, S. Naz, M. Z. Afzal, S. F. Rashid, M. Liwicki, and A. Dengel, "KHATT: A Deep Learning Benchmark on Arabic Script," *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, vol. 7, pp. 10–14, 2018, doi: 10.1109/ICDAR.2017.358.
[21] R. Ahmad, S. Naz, M. Afzal, S. Rashid, M. Liwicki, and A. Dengel, "A deep learning based arabic script recognition system: Benchmark on khat," *Int. Arab J. Inf. Technol.*, vol. 17, no. 3, pp. 299–305, 2020, doi: 10.34028/iajit/17/3/3.
[22] L. Gui, X. Liang, X. Chang, and A. G. Hauptmann, "Adaptive context-aware reinforced agent for handwritten text recognition," *Br. Mach. Vis. Conf. 2018, BMVC 2018*, pp. 1–13, 2019.
[23] M. Yousef, K. F. Hussain, and U. S. Mohammed, "Accurate, data-efficient, unconstrained text recognition with convolutional neural networks," *Pattern Recognit.*, vol. 108, no. 8, pp. 1–13, 2020, doi: 10.1016/j.patcog.2020.107482.
[24] S. K. Jemni, Y. Kessentini, S. Kanoun, and J. M. Ogier, "Offline Arabic handwriting

Power System Technology
ISSN:1000-3673
Received: 5-2-2024 Revised: 15-2-2024 Accepted: 17-2-2024

recognition using blstms combination," *Proc. - 13th IAPR Int. Work. Doc. Anal. Syst. DAS 2018*, pp. 31–36, 2018, doi: 10.1109/DAS.2018.54.

[25] S. K. Jemni, Y. Kessentini, and S. Kanoun, "Improving Recurrent Neural Networks for Offline Arabic Handwriting Recognition by Combining Different Language Models," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 34, no. 12, p. 2052007, Nov. 2020, doi: 10.1142/S0218001420520072.

[26] S. K. Jemni, Y. Kessentini, and S. Kanoun, "Out of vocabulary word detection and recovery in Arabic handwritten text recognition," *Pattern Recognit.*, vol. 93, pp. 507–520, 2019, doi: 10.1016/j.patcog.2019.05.003.

[27] Z. Noubigh, A. Mezghani, and M. Kherallah, "Contribution on Arabic Handwriting Recognition Using Deep Neural Network," 2021, pp. 123–133. doi: 10.1007/978-3-030-49336-3_13.

[28] Z. Noubigh, A. Mezghani, and M. Kherallah, "Open Vocabulary Recognition of Offline Arabic Handwriting Text Based on Deep Learning," in *20th International Conference on Intelligent Systems Design and Applications (ISDA 2020)*, 2021, pp. 92–106. doi: 10.1007/978-3-030-71187-0_9.

[29] Z. Noubigh, A. Mezghani, and M. Kherallah, "Transfer learning to improve arabic handwriting text recognition," *Proc. - 2020 21st Int. Arab Conf. Inf. Technol. ACIT 2020*, p. TransferLlearning to improve Arabic handwriting te, 2020, doi: 10.1109/ACIT50332.2020.9300105.

[30] Z. Noubigh, A. Mezghani, and M. Kherallah, "Densely connected layer to improve VGGnet-based CRNN for Arabic handwriting text line recognition," *Int. J. Hybrid Intell. Syst.*, vol. 17, no. 3–4, pp. 113–127, 2021, doi: 10.3233/his-210009.

[31] T. B. A. Gader and A. K. Echi, "Attention-based CNN-ConvLSTM for Handwritten Arabic Word Extraction," *Electron. Lett. Comput. Vis. Image Anal.*, vol. 21, no. 1, pp. 121–134, 2022, doi: 10.5565/rev/elcvia.1433.

[32] S. A. Mahmoud *et al.*, "KHATT: An open Arabic offline handwritten text database," *Pattern Recognit.*, vol. 47, no. 3, pp. 1096–1112, 2014, doi: 10.1016/j.patcog.2013.08.009.

[33] S. A. Mahmoud *et al.*, "KHATT: Arabic offline Handwritten Text Database," *Proc. - Int. Work. Front. Handwrit. Recognition, IWFHR*, no. January, pp. 449–454, 2012, doi: 10.1109/ICFHR.2012.224.

[34] S. K. Jemni, S. Ammar, and Y. Kessentini, "Domain and writer adaptation of offline Arabic handwriting recognition using deep neural networks," *Neural Comput. Appl.*, vol. 34, no. 3, pp. 2055–2071, 2022, doi: 10.1007/s00521-021-06520-7.

[35] M. F. Benzeghiba, "A Comparative Study on Optical Modeling Units for Off-Line Arabic Text Recognition," *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, vol. 1, pp. 1025–1030, 2017, doi: 10.1109/ICDAR.2017.170.

[36] F. Slimane *et al.*, "ICFHR2014 Competition on Arabic Writer Identification Using AHTID/MW and KHATT Databases," *Proc. Int. Conf. Front. Handwrit. Recognition, ICFHR*, vol. 2014-Decem, pp. 797–802, 2014, doi: 10.1109/ICFHR.2014.139.

[37] Tzutalin, "labelImg," 2015. https://github.com/tzutalin/labelImg

[38] J. Yang, P. Ren, and X. Kong, "Handwriting Text Recognition Based on Faster R-CNN," *Proc. - 2019 Chinese Autom. Congr. CAC 2019*, pp. 2450–2454, 2019, doi:

Power System Technology
ISSN:1000-3673
Received: 5-2-2024 Revised: 15-2-2024 Accepted: 17-2-2024

10.1109/CAC48633.2019.8997382.
[39] T. Nazir, A. Irtaza, J. Rashid, M. Nawaz, and T. Mehmood, "Diabetic Retinopathy Lesions Detection using Faster-RCNN from retinal images," *Proc. - 2020 1st Int. Conf. Smart Syst. Emerg. Technol. SMART-TECH 2020*, no. December, pp. 38–42, 2020, doi: 10.1109/SMART-TECH49988.2020.00025.
[40] C. Huang, A. Yu, and H. He, "Using combined Soft-NMS algorithm Method with Faster R-CNN model for skin lesion detection," *ACM Int. Conf. Proceeding Ser.*, pp. 5–8, 2020, doi: 10.1145/3449301.3449303.
[41] S. Albahli, M. Nawaz, A. Javed, and A. Irtaza, "An improved faster-RCNN model for handwritten character recognition," *Arab. J. Sci. Eng.*, vol. 46, no. 9, pp. 8509–8523, 2021, doi: 10.1007/s13369-021-05471-4.
[42] P. Sharma, S. Gupta, S. Vyas, and M. Shabaz, "Retracted: Object detection and recognition using deep learning-based techniques," *IET Commun.*, vol. 17, no. 13, pp. 1589–1599, 2023, doi: 10.1049/cmu2.12513.
[43] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.
[44] R. Gyawali and D. Raj, "An Approach for the Employee Face Recognition by RPN and Faster R-CNN Techniques," pp. 231–237, 2019.
[45] Y. Zhang, Y. Chen, C. Huang, and M. Gao, "Object detection network based on feature fusion and attention mechanism," *Futur. Internet*, vol. 11, no. 1, pp. 1–14, 2019, doi: 10.3390/fi11010009.
[46] M. Lu, Y. Mou, C. L. Chen, and Q. Tang, "An efficient text detection model for street signs," *Appl. Sci.*, vol. 11, no. 13, 2021, doi: 10.3390/app11135962.
[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.
[48] H. Zhao *et al.*, "Identification Method for Cone Yarn Based on the Improved Faster R-CNN Model," *Processes*, vol. 10, no. 4, 2022, doi: 10.3390/pr10040634.
[49] L. Du *et al.*, "A Novel Object Detection Model Based on Faster R-CNN for Spodoptera frugiperda According to Feeding Trace of Corn Leaves," *Agric.*, vol. 12, no. 2, 2022, doi: 10.3390/agriculture12020248.
[50] H. Nguyen, "Improving Faster R-CNN Framework for Fast Vehicle Detection," *Math. Probl. Eng.*, vol. 2019, 2019, doi: 10.1155/2019/3808064.
[51] 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, 2022, doi: 10.1155/2022/7812410.
[52] C. Cao *et al.*, "An Improved Faster R-CNN for Small Object Detection," *IEEE Access*, vol. 7, no. April, pp. 106838–106846, 2019, doi: 10.1109/ACCESS.2019.2932731.
[53] R. A. Alawwad, O. Bchir, and M. M. Ben Ismail, "Arabic Sign Language Recognition using Faster R-CNN," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 3, pp. 692–700, 2021, doi: 10.14569/IJACSA.2021.0120380.
[54] A. A and L. K, "A Faster RCNN Based Image Text Detection and Text to Speech Conversion," *Int. J. Electron. Commun. Eng.*, vol. 5, no. 5, pp. 11–14, 2018, doi:

Power System Technology
ISSN:1000-3673
Received: 5-2-2024 Revised: 15-2-2024 Accepted: 17-2-2024

10.14445/23488549/ijece-v5i5p103.

[55] J. Deng, Y. Lu, and V. C. S. Lee, "Concrete crack detection with handwriting script interferences using faster region-based convolutional neural network," *Comput. Civ. Infrastruct. Eng.*, vol. 35, no. 4, pp. 373–388, 2020, doi: 10.1111/mice.12497.

[56] Y. Song, Q. K. Pan, L. Gao, and B. Zhang, "Improved non-maximum suppression for object detection using harmony search algorithm," *Appl. Soft Comput. J.*, vol. 81, p. 105478, 2019, doi: 10.1016/j.asoc.2019.05.005.

[57] N. S. Ouf, "Leguminous seeds detection based on convolutional neural networks: Comparison of Faster R-CNN and YOLOv4 on a small custom dataset," *Artif. Intell. Agric.*, vol. 8, pp. 30–45, 2023, doi: 10.1016/j.aiia.2023.03.002.

[58] Jiangqiao Y, et al. IoU-adaptive deformable R-CNN: Make full use of IoU for multi-class object detection in remote sensing imagery. *Remote Sensing*. 2019; 11(3), 256, pp. 1-22.

[59] Colin D, Rufin V, Didier S, Thomas O. DAROD: A Deep Automotive Radar Object Detector on Range-Doppler maps. In: Proceeding of the 2022 IEEE Intelligent Vehicles Symposium (IV). 2022; pp. 112–118.