ArabicWordNet: fine-tuning MobileNetV2-based model for Arabic handwritten words recognition

Hakim A. Abdo[1,2], Ahmed Abdu[3], Ramesh R. Manza[1], Shobha Bawiskar[4]

[1]Department of Computer Science and IT, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, India
[2]Department of Computer Science, Hodeidah University, Al-Hudaydah, Yemen
[3]Department of Software Engineering, Northwestern Polytechnical University, Xi'an, China
[4]Department of Digital and Cyber Forensics, Government Institute of Forensic Science, Aurangabad, India

---

ABSTRACT- Recognizing Arabic handwritten words is challenging. The variability in handwriting styles and the abundance of vocabulary for Arabic pose significant challenges. In this work, we propose a deep learning model that leverages transfer learning and undergoes an ablation study for improved performance. We evaluate five pre-trained architectures, namely DenseNet201, MobileNetV2, ResNet50, InceptionV3, and VGG16, and show that MobileNetV2 emerges as the best performer with high validation accuracy of 96.93%. We further enhanced the proposed model through the ablation study, achieving a validation accuracy of 98.98% and a test accuracy of 99.02%. Our results demonstrate the effectiveness of our approach in recognizing Arabic handwritten words, making it a valuable contribution to the field of Arabic handwriting recognition for cheque processing.

Keyword: Arabic Handwritten Word Recognition, Transfer Learning, Fine-Tuned Mobilenetv2, Ablation Study.

1. INTRODUCTION The Arabic writing system utilizes a uniquely cursive handwritten script for composing words and sentences. However, studies [1] have shown that it exhibits considerable stylistic variation at the individual level due to factors like differences in education, cultural background, and personal preferences in handwriting style. Additionally, Arabic has several formal calligraphic variants such as Naskh, Kufi, and Thuluth [2] scripts, each with their own visually distinct features and strokes. Another area of complexity is numerals and numerical amounts expressed in the Arabic script, which are commonly used in vital documentation like financial cheques and transactions in Arabic-speaking countries. These numeric expressions have right-to-left reading order conventions [3] that add difficulties for recognition systems.

Optical character recognition pipelines comprise several key stages: preprocessing, segmentation, and recognition [4]. Preprocessing techniques like binarization, smoothing, rotation corrections aim to clean up the input image and improve quality to simplify later pipeline stages. Segmentation then isolates individual component letters or words from the preprocessed input images based on visual patterns. Finally, recognition assigns textual or numerical labels to each segmented component by matching to known templates or features. However, the cursive flowing nature of Arabic script poses challenges for accurate segmentation and recognition compared to more separated scripts. Still, advancing Arabic optical character recognition remains an active area of research to enhance automation for practical applications like mail sorting, office document handling, bank cheque processing and other financial transactions.

Experts in the field [5] emphasize that the overall accuracy of Arabic handwriting recognition systems relies heavily on the specific choices and implementations made in the preprocessing, feature extraction, and classification stages. Feature extraction aims to mathematically represent the visual qualities and patterns in the Arabic script components in ways that can simplify the overall classification challenge. Numerical legal amounts expressed in words on financial cheques pose additional difficulties, as discussed by Khorsheed et al. [6]. Challenges include the extensive Arabic vocabulary for expressing numbers, interpretive complexities, informal language usage, spelling variations, mistakes, which all hamper recognition accuracy.

A survey of past research methods for tackling Arabic script recognition shows employment of varied techniques like statistical patterns, syntactic rules, Fourier and contour shape analyses paired with conventional classification algorithms. However, recently deep learning approaches have shown tremendous promise for automating large-scale feature extraction from visual data[7], reducing the need for manual human-crafted features. Additionally, fine-tuning pretrained deep neural networks by leveraging transfer learning mitigates challenges posed by limited dataset sizes and also boosts performance on tackling complex Arabic recognition tasks[7]. In summary, advancing Arabic optical character recognition requires developing specialized algorithms tailored to model the formal intricacies and challenges presented by its cursive script nature. This research proposes a deep learning model based on transfer learning for recognising Arabic handwritten legal amount words. The steps involved in recognizing Arabic handwritten words are presented in Figure 1 in this research. To develop the model, five pre-trained and fine-tuned transfer learning architectures are evaluated, including VGG16, MobileNetV2, ResNet50, DenseNet201, and InceptionV3, to identify the best-performing model. Each pre-trained model was fine-tuned by retaining the original weights of the primary layers and retraining the last layers using AHBD dataset.

This paper's contributions are:

- The study presents the evaluation of various pre-trained DL models on the AHDB dataset and presents the ArabicWordNet model based on the fine-tuning with ablation study of the best-performing model. The results demonstrate the effectiveness of the proposed ArabicWordNet model on the AHDB dataset, with high accuracy and low bias towards any one class.

In summary, the rest of this paper is structured as follows: Section 2 covers the background information and related studies, Section 3 details the methods and materials used, Section 4 outlines the experimental setup, and finally, Section 5 presents and discusses the results obtained.

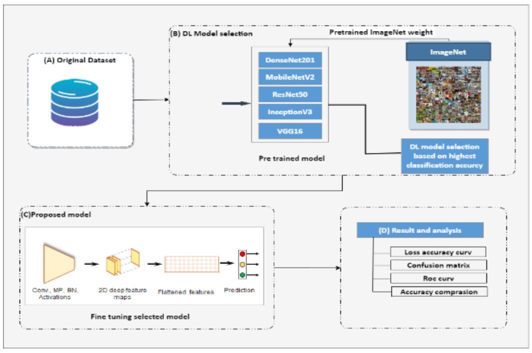


Figure 1 Workflow of the recognition process, the original AHDB dataset is represented by block (A). Block (B) displays the experimentation with various pre-trained models, and the selection of the model with the highest accuracy on the AHDB dataset, leading to the creation of the proposed ArabicWordNet model. Block (C) depicts the results of an ablation study on the proposed model. Block (D) illustrates the analysis of the final model's results using performance metrics.

2.Background and Related work

2.1 Transfer Learning

Transfer Learning (TL) utilizes pre-trained convolutional neural networks like VGG16, DenseNet201, ResNet50, MobileNetV2, and InceptionV3 to save computation time and enhance performance on new classification tasks with limited labeled training data [8]. The key idea is to transfer knowledge from the pre-trained models to the new model.

- **VGG16**

VGG16 is a deep convolutional neural network introduced by Simonyan and Zisserman [9]. It was trained on the ImageNet dataset achieving high accuracy. VGG16 is characterized by its depth, enabling it to learn complex features. Its architecture includes 13 Conv layers (C1 to C13) and three FC layers (FC1 to FC3).

- **DeneNet201**

DenseNet [10] was designed to improve gradient propagation through identity mapping. By linking all layers, higher level feature maps can express all data, allowing the network to achieve higher accuracy with fewer parameters.

- **ResNet50**

ResNet50 utilizes combinations of small and large convolutional filters addressing performance degradation in deep CNNs [11]. Its architecture, with over 23 million trainable parameters, consists of convolutional and pooling layers.

- **MobileNetV2**

MobileNetV2 is a convolutional neural network proposed by the Google community; it uses depthwise separable convolutions, which are more computationally efficient than traditional convolutions, to reduce the number of parameters and improve the speed of the network. Additionally, it uses a technique called linear bottlenecks to further reduce the number of parameters and increase the efficiency of the network[12].

- **InceptionV3**

InceptionV3 employs factorization, regularization, and parallelization to significantly reduce computational costs compared to previous Inception models [13]. Key architecture changes include label smoothing, factorized 7x7 convolutional layers, and an auxiliary classifier to pass label information through the network. Smaller convolutions also reduce training time.

2.2 Related Work

Several studies have employed different techniques for recognizing Arabic handwritten text. A common focus has been using hand-selected feature extraction methods. For example, Souici et al. [14] developed a hybrid neural network and symbolic classifier that analyzed contours in images for attributes like loop count to recognize words. Their method was tested on 1200 words from 25 writers. Hassen et al. .[15] also created a recognition system using statistical feature extraction methods like histograms of oriented gradients, invariant moments, and Gabor filters. These features were fed into classifiers like sequential minimal optimization. Meanwhile, Al-Nuzaili et al. [16] specifically focused on Gabor filter features at two different image sizes. They compared multiple classifiers as well, including extreme learning machines. Both research teams evaluated performance on datasets like AHDB and CENPARMI.

Overall, a range of techniques have shown promise for recognizing Arabic script, including legal amounts. Hand-crafted feature extraction is common, though researchers are testing neural networks, too. Comparing methods on standard datasets has helped benchmark progress. There are opportunities to build on these works by exploring additional recognition frameworks and testing more script samples.

Deep learning models like convolutional neural networks (CNNs) are achieving state-of-the-art results on many pattern recognition tasks, including recent advances in Arabic handwriting recognition[17]–[19].. However, as Altwaijry et al. [20] discuss, deep learning has yet to be extensively explored for Arabic script. Their 2018 paper proposed a CNN for recognising Arabic words, testing it on datasets like AHCD and achieving 97% accuracy. Around the same time, Maalej et al. [21] developed a hybrid CNN and long short-term memory network (LSTM) system that used the CNN to extract features and the LSTM to classify letter sequences, aided by a connectionist temporal classification (CTC) layer. In another promising approach, Elleuch et al. [22] paired a CNN for feature learning with a support vector machine (SVM) classifier, evaluating performance on Arabic handwriting datasets HACDB and IFN/ENIT.

More recent studies have continued to advance Arabic handwriting recognition with deep learning. For example, El-Melegy et al. [23] applied a CNN to recognize Arabic monetary amounts in the AHDB cheque dataset specifically. Meanwhile, Korichi [24] benchmarked various CNN architectures, including VGG-16 and ResNets, for amount recognition. They found that ResNets achieved over 98% accuracy and that techniques like data augmentation and dropout improved generalizability.

3. Methods and Materials –

This section details the proposed Arabic handwritten legal amount words recognition model named ArabicWordNet. The overall architecture of the proposed model is shown in Figure 3. The following subsections will provide a comprehensive dataset preparation overview, the method of developing the proposed model, and how its performance was improved.

3.1 AHDB dataset-

The AHDB dataset [25] is a publicly available resource that contains 3,043 Arabic common word images distributed across 30 distinct classes, making it a valuable repository for Arabic handwritten word recognition research. The dataset is unique in that it comprises handwritten words and texts from 100 different individuals, adding complexity and variability to the dataset. The AHDB dataset is used as a benchmark for model assessment in this research. The dataset includes the most popular words in Arabic writing. The AHDB dataset has undergone preprocessing operations such as slant correction, thinning, and width normalization to make the task of recognition simpler. The AHDB dataset is designed to provide a training and testing set for Arabic text recognition research.

3.2 The proposed model (ArabicWordNet)

In this study, we conducted several experiments to identify the best transfer learning model for the classification of legal amount words; we tried out five pre-trained convolutional neural networks: VGG16, DeneNet201, ResNet50, MobileNetV2, and InceptionV3 to identify the best model depending on accuracy. throughout the training phase, just the weights of the final Fully Connected layer were modified and updated across all models. We developed this transfer learning method by adding a single flatten and dense layer at the end of the pre-trained models. To fine-tune the models, a flatten layer is added before the FC layers, and the output layer is modified with 33 neurons which is the number of our classes. The fine-tuned MobileNetV2 architecture gives the highest classification accuracy of the five abovementioned architectures. Therefore, we propose a model called ArabicWordNet that is based on a tweaked MobileNetV2 architecture and for which we conduct several experiments on our dataset to evaluate its performance. Figure 4 depicts the model's architecture. In the fine-tuned MobileNetV2 model, an image with the dimensions (224, 224, 3) is used as input, and the model is comprised of 17 bottleneck depthwise blocks and one Conv2d layer with reLU6: Each block has three components: a 1x1 convolution, a 3x3 depthwise convolution, and another 1x1 convolution. The first 1x1 convolution is used to reduce the number of input channels, the second 3x3 depthwise convolution performs the actual depthwise convolution, and the final 1x1 convolution is used to increase the number of output channels. A ReLU6 activation function that applies a non-linear activation function to the output of the blocks.

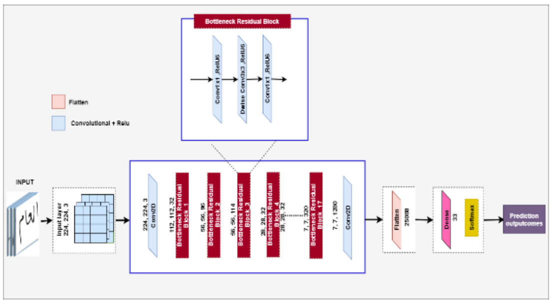


Figure 2 Architecture of ArabicWordNet model, a fine-tuned MobileNetV2 network

To build our ArabicWordNet model, we modified the MobileNetV2 model by adding flatten layer and a dense layer behind the model's last convolutional layers. The input layer of our model is 224*224*3, which is an image size of 224*224 with RGB. Within this context, all 17 bottleneck depthwise blocks consist of a 1x1 convolution, a 3x3 depthwise convolution, and another 1x1 convolution with different size channels. The final Conv2D layer produces an output feature map with a size of 7 * 7 * 1280 after stacking convolutional and reLU. A flatten layer has been added to flatten the 2-dimension feature map to 1 * 25,088 feature vector. In addition, a 33-channel dense layer has been added for the 33 classes. An endpoint Softmax activation function is used to normalize the FC-obtained classification vector. As clarified above, our proposed ArabicWordNet model is created by adding a flatten layer and dense layer succeeding the 17th bottleneck depthwise block of the MobileNetV2 architecture. As indicated in figure 5, our experiment has two stages: Pre-trained and fine-tuning. Throughout the training process in the pre-trained stage, the layers' weights keep freezing. In the fine-tuning stage, the green boxes indicate flatten and dense layers of ArabicWordNet that are being trained

1.1. Parameters Setting

The ablation study on the proposed ArabicWordNet model revealed that utilizing the GlobalAveragePooling layer, optimal batch size, loss function, and optimizer improves classification accuracy. Table 3 summarizes the final configuration parameters for the ArabicWordNet model. During the training phase for this final model, the accuracy and loss for both training and validation were recorded, and a confusion matrix was calculated.

with new features of our Arabic handwritten legal amount words. The final fully connected layer is adjusted to classify 33 classes in place of 1000 classes. Thus, we have retrained the MobileNetV2 model using AHDB dataset after transferring its pre-trained weights using transfer learning. Finally, the probabilities for each class are computed using the Softmax activation function and then yield the projected result.

3.3 Training method

To determine the optimal weights for mapping inputs to outputs in our dataset, we trained the models for 30 epochs with a batch size of 8. The Adam optimization algorithm was used with a default learning rate of 0.001. The optimal model's weights were saved by recording the minimum loss value achieved during training using the Keras callback function.

3.4 Experimental setup

This section provides the specific settings for the evaluation experiments. The experiments were conducted on a CPU Intel® Core™ i7 with NVIDIA GeForce MX250 and CUDA 11.2. The implementation was done in Python 3.9.6 using Tensorflow 2.5.0 and Keras 2.5.0..

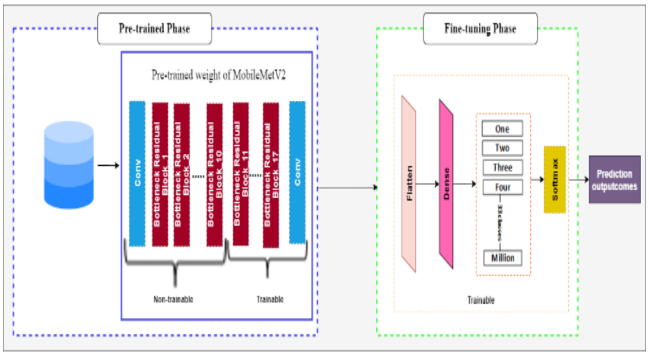
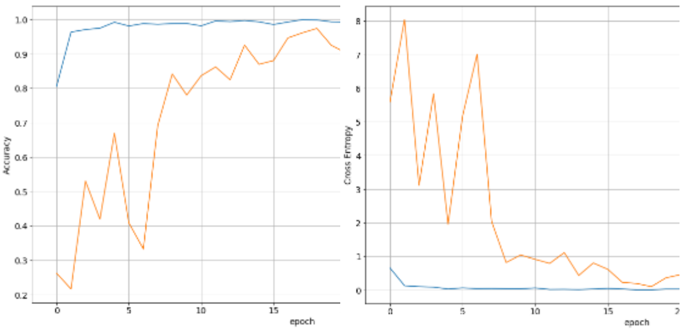


Figure 3 fine-tuning process.

Table 1 Proposed architecture configuration after ablation study.

Parameter	Value
Input size	224 × 224
Epochs	90
function of optimization	Adam
Learning rate	0.001
Batch size	32
Function of activation	Softmax

Figure 6 shows our best-performing model's learning curves for loss and accuracy. The training curve shows smooth and steady progress from the first to the final epoch. The validation and training accuracy curves do not offer a significant gap, indicating the absence of overfitting.



(a) (b)

Figure 4 Accuracy and loss curves of the model during training: (a) Accuracy curves;(b) loss curves.

The confusion matrix for the most accurate model is shown in figure 7. The test images' actual label is represented by the row values, while the column values represent their predicted label. The TP is the diagonal value. The proposed ArabicWordNet model's confusion matrix predicts all classes almost equally and has the optimal configuration based on Adam's ablation research and learning rate of 0.0008. However, the model had the best results for most classes.

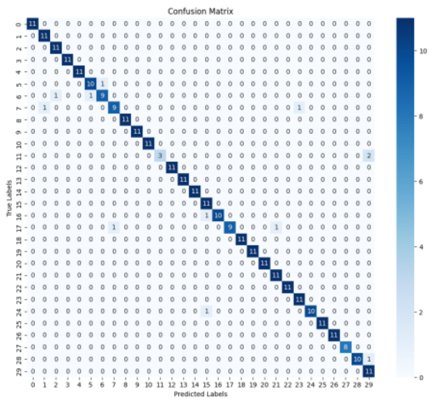


Figure 5 confusion matrix of the proposed model

1.2. Performance evaluation metrics

To assess the transfer learning models' performance, we employed a range of metrics: accuracy (ACC), recall, precision, and F1-score[26]. For each model, we calculated a Confusion Matrix. From the Confusion Matrix, we obtained the values for false negatives (FN), false positives (FP), true negatives (TN), and true positives (TP).

$$ACC = \frac{TP + TN}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \quad (1)$$

$$Recall = \frac{TP}{\mathrm{TP} + \mathrm{FN}} \quad (2)$$

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$\mathrm{F}_1 = 2\,\frac{Precision * Recall}{Precision + Recall} \quad (4)$$

4. Results and Discussion

The results and discussion are guided by the following research questions:

RQ1: How does MobileNetV2 outperform other transfer learning models on our AHDB dataset?

RQ 2: How is ArabicWordNet compared with the state-of-the-art recognising Arabic handwriting legal amount words approaches?

RQ3: How do external parameters affect the performance of ArabicWordNet?

RQ1

To determine the most effective transfer learning model for classifying legal amount words, we evaluated five state-of-the-art classification models (VGG16, DenseNet201, ResNet50, MobileNetV2, and InceptionV3) based on their accuracy. During the training phase, we only adjusted and updated the weights of the final fully connected layer across all models. We evaluated the performance of all five models on our AHDB dataset through experiments. The outcome of the experiments, as presented in Table 4, revealed that the MobileNetV2 model performed the best with a training accuracy of 99.20% and a validation accuracy of 96.93% while keeping the training loss at 0.0422 and validation loss at 0.1957. The other models: VGG16, DenseNet201, and InceptionV3 had training accuracy of 99.18%, 99.14% and 99.05%, respectively, which were comparable to that of the best-performing model. Similarly, the validation accuracy for VGG16, DenseNet201 and InceptionV3 were 93.90%, 91.59%, and 88.68%, respectively. Additionally, the recall, precision, and F1 score for MobileNetV2 were 97.14%, 97.14%, and 97.15%, respectively, and it had a high sensitivity of 98.03% and a specificity of 98.14%. These results demonstrate the strong classification ability and performance of MobileNetV2 on our AHDB dataset. Consequently, MobileNetV2 was selected as the base model and further ablation studies were conducted to enhance its performance for classification tasksTable 2 Comparison of the performance of transfer learning models.

Models	F1score	Precision	Recall	Tr_Acc	Val_Acc	Tr_Loss	Val_Loss	Ts_Acc	Ts_loss
VGG16	93.	93.	93.	99.18	93.90	3.738e-06	0.41447	90.08	0.3557
MobileNetV2	97.15	97.14	97.14	99.20	96.93	.000422	.1957	97.14	0.1684
ResNet50	86.	87.	86.	93.94	75.61	.19032	1.0158	76.78	1.0158
DeneNet201	91.61	92.	91.61	99.14	91.59	.001195	2.7561	91.61	2.782
InceptionV3	89.	8.	89.	99.05	88.68	.02239	.4387	89.72	0.3975

RQ2

In this section, we compare our model with recent studies discussed in the literature review by presenting a comparison in Table 5.

Table 3 Comparison proposed model with existing studies.

Study	Dataset	Accuracy (%)
[28]	AHDB	83.06%
[15]	AHDB	91.59%
[16]	AHDB	72.79% and 89.29% for ELM and SMO classifiers, respectively.
	CENPARMI	80.86% and 86.72% for ELM and SMO classifiers, respectively.
[29]	IFN-ENIT	87%.
[23]	AHDB	97.85%
[24]	AHDB	98.5%
Our ArabicWordNet model	AHDB	99.02%.

As seen in the table above, most studies have employed handcrafted-based methods, while only two have utilized CNN architectures. Compared to handcrafted-based methods, CNN architectures have proven their effectiveness in recognizing literal amounts of Arabic handwriting images. The high accuracy achieved by our proposed model illustrates that transfer learning can be an effective approach when working with a limited number of images and that fine-tuning the model's architecture and hyper-parameters through an ablation study can greatly impact the overall accuracy.

RQ3

We performed an ablation study to understand the effect of different components of the proposed ArabicWordNet model on its performance. An ablation study involves altering or removing specific parts of a model and observing the impact on its performance; this is a useful approach for determining which components are crucial for the model's performance and for fine-tuning the model to achieve optimal performance. In the case of the ArabicWordNet model, we conducted experiments that modified the batch size, flatten layer, loss function, optimizer, and learning rate, to understand the effect of these components on the model's performance. Through this ablation study, we gained insight into which model components have the most significant impact on improving its robustness and accuracy.

Altering Flatten Layer

We modified our model's architecture by replacing the flatten layer with either a GlobalAveragePooling2D layer or a GlobalMaxPooling2D layer and observed the impact on the model's performance. In a convolutional neural network (CNN), the flatten layer is used to convert the 2-dimensional feature maps output by the preceding convolutional layers into a 1-dimensional array, which can be input into a fully connected (FC) layer. The GlobalAveragePooling2D and GlobalMaxPooling2D layers are similar to the flatten layer in that they also convert 2D feature maps into 1D arrays, but they do so use different techniques. The GlobalAveragePooling2D layer takes the average of all the values in the feature map, while the GlobalMaxPooling2D layer takes the maximum value. Table 6 illustrates that for the GlobalAveragePooling2D layer, the network's performance improved significantly in terms of accuracy. On the other hand, when using the GlobalMaxPooling2D layer, the network's performance increases in accuracy.

Table 4 Results of ablation study on altering Flatten Layer

Ablation Study case	Layer name	Ts_Acc	Tr_Acc	Ts_Loss	Tr_Loss (%)	outcomes
Altering Layer	Flatten	97.14	99.20	0.1684	0.000422	
	GlobalAveragePooling2D	98.92	1	0.0699	0.000258	improved in accuracy
	GlobalMaxPooling2D	98.61	99.74	0.1612	0.000322	improved in accuracy

Altering Batch Size

The batch size is an essential hyperparameter when training a deep learning model, as it can significantly influence its performance. In our study, we examined the impact of varying batch sizes on the performance of our proposed model. By adjusting the batch size from 8 to 16 and 32, we observed an enhancement in both test and validation accuracy. Table 7 illustrates that for the highest test accuracy of 98.02% and the lowest loss of 0.0474 were recorded with a batch size of 32. Consequently, we decided to use this batch size for further experimentation.

Table 5 Results of ablation study on altering batch Size.

Ablation Study case	Batch Size	Ts_Acc	Tr_Acc	Ts_Loss	Tr_Loss (%)	outcomes
Altering Batch Size	8	98.92	100	0.0699	0.000258	
	16	98.93	100	0.0680	0.000201	improved in accuracy
	32	99.02	100	0.0474	0.000152	improved in accuracy

Altering Loss Functions:

We conducted experiments using various loss functions, such as Categorical Crossentropy, Cosine Similarity, and Mean Squared Error to determine the optimal loss function for our proposed models. The results of the model's performance with the selected loss functions are presented in Table 5. The model achieved the highest test accuracy of 99.02% when using the Categorical Crossentropy loss function. The test accuracy slightly decreased for both the Cosine Similarity (97.59%) and Mean Squared Error loss functions (98.16%). Based on these results, the Categorical Crossentropy loss function was selected for further testing to achieve the highest classification performance.

Table 6 Results of ablation study on altering loss Functions.

Ablation Study case	Loss Function	Ts_Acc	Tr_Acc	Ts_Loss	Tr_Loss	outcomes
Altering Loss Functions	Categorical Crossentropy	99.02	100	0.0680	0.000152	
	Cosine similarity	97.59	99.80	0.1684	0.000522	Accuracy dropped
	Mean Squared Error	98.16	99.95	0.1516	0.000412	Accuracy dropped

4. CONCLUSIONS:

The present study aimed to determine the best pretrained CNN model for Arabic handwritten legal amount recognition using the AHDB dataset. Five pre-trained CNN models were experimented with, and it was found that the MobileNetV2 model yielded the highest accuracy. An ArabicWordNet model was proposed, using MobileNetV2 as its foundation, and an ablation study was conducted to evaluate and improve the model's robustness. The proposed ArabicWordNet model performed best with optimiser Adam and a learning rate of 0.0008, resulting in a training accuracy of 100%, a validation accuracy of 98.98%, and a test accuracy of 99.02%. The findings of this study demonstrate that transfer learning can be an appropriate approach when working with a small number of images and that configuring the model's architecture and hyper-parameters using an ablation study can significantly impact the overall accuracy. This study also highlights the importance of creating large and diverse datasets for Arabic handwritten legal amount recognition tasks, as it is crucial for achieving high accuracy and robustness in practical applications.

References

[1] S. Djaghbellou, A. Bouziane, A. Attia, and Z. Akhtar, "A Survey on Arabic Handwritten Script Recognition Systems," *Int. J. Artif. Intell. Mach. Learn.*, vol. 11, no. 2, pp. 1–17, 2021, doi: 10.4018/ijaiml.20210701.oa9.

[2] F. Alkhateeb, M. Barhoush, and E. Al-abdallah, "Developing a System for Converting a Numeral Text into a Digit Number : Abacus Application," vol. 25, no. 4, pp. 611–628, 2016, doi: 10.1515/jisys-2015-0029.

[3] H. A. Abdo, A. Abdu, R. R. Manza, and S. Bawiskar, "An approach to analysis of arabic text documents into text lines, words, and characters," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 26, no. 2, pp. 754–763, 2022, doi: 10.11591/ijeecs.v26.i2.pp754-763.

[4] I. Ahmad and S. A. Mahmoud, "Arabic bank check processing: State of the art," *J. Comput. Sci. Technol.*, vol. 28, no. 2, pp. 285–299, 2013, doi: 10.1007/s11390-013-1332-6.

[5] A. Patil and M. Rane, "Convolutional Neural Networks: An Overview and Its Applications in Pattern Recognition," *Smart Innov. Syst. Technol.*, vol. 195, pp. 21–30, 2021, doi: 10.1007/978-981-15-7078-0_3.

[6] Q. A. Al-Nuzaili, S. Z. M. Hashim, F. Saeed, M. S. Khalil, and D. Mohamad, "Pixel distribution-based features for offline Arabic handwritten word recognition," *Int. J. Comput. Vis. Robot.*, vol. 7, no. 2, pp. 99–122, 2017.

[7] S. Faizullah, M. S. Ayub, S. Hussain, and M. A. Khan, "A Survey of OCR in Arabic Language: Applications, Techniques, and Challenges," *Appl. Sci. 2023, Vol. 13, Page 4584*, vol. 13, no. 7, p. 4584, Apr. 2023, doi: 10.3390/APP13074584.

[8] F. Zhuang *et al.*, "A Comprehensive Survey on Transfer Learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, 2021, doi: 10.1109/JPROC.2020.3004555.

[9] 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.

[10] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2261–2269, 2017, doi: 10.1109/CVPR.2017.243.

[11] 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.

[12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 4510–4520, 2018, doi: 10.1109/CVPR.2018.00474.

[13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 2818–2826, 2016, doi: 10.1109/CVPR.2016.308.

[14] L. Souici-Meslati and M. Sellami, "A hybrid approach for arabic literal amounts recognition," *Arab. J. Sci. Eng.*, vol. 29, no. 2 B, pp. 177–194, 2004.

[15] H. Hassen and S. Al-Maadeed, "Arabic handwriting recognition using sequential minimal optimization," in *1st IEEE International Workshop on Arabic Script Analysis and Recognition, ASAR 2017*, 2017. doi: 10.1109/ASAR.2017.8067764.

[16] Q. Al-Nuzaili, S. Al-Maadeed, H. Hassen, and A. Hamdi, "Arabic Bank Cheque Words Recognition Using Gabor Features," *2nd IEEE Int. Work. Arab. Deriv. Scr. Anal. Recognition, ASAR 2018*, pp. 84–89, 2018, doi: 10.1109/ASAR.2018.8480197.

[17] A. A. Alani, "Arabic handwritten digit recognition based on restricted Boltzmann machine and convolutional neural networks," *Inf.*, vol. 8, no. 4, 2017, doi: 10.3390/info8040142.

[18] 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*, no. January, pp. 182–188, 2021, doi: 10.1109/ICICS52457.2021.9464596.

[19] B. H. Nayef, S. N. H. S. Abdullah, R. Sulaiman, and Z. A. A. Alyasseri, "Optimized leaky ReLU for handwritten Arabic character recognition using convolution neural networks," *Multimed. Tools Appl.*, vol. 81, no. 2, pp. 2065–2094, 2022, doi: 10.1007/s11042-021-11593-6.

[20] N. Altwaijry and I. Al-Turaiki, "Arabic handwriting recognition system using convolutional neural network," *Neural Comput. Appl.*, vol. 33, no. 7, 2021, doi: 10.1007/s00521-020-05070-8.

[21] R. Maalej and M. Kherallah, "Convolutional Neural Network and BLSTM for Offline Arabic Handwriting Recognition," in *ACIT 2018 - 19th International Arab Conference on Information Technology*, 2019. doi: 10.1109/ACIT.2018.8672667.

[22] M. Elleuch, R. Maalej, and M. Kherallah, "A New design based-SVM of the CNN classifier architecture with dropout for offline Arabic handwritten recognition," in *Procedia Computer Science*, 2016. doi: 10.1016/j.procs.2016.05.512.

[23] M. El-Melegy, A. Abdelbaset, A. Abdel-Hakim, and G. El-Sayed, "Recognition of Arabic Handwritten Literal Amounts Using Deep Convolutional Neural

Networks," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11868 LNCS, pp. 169–176, 2019, doi: 10.1007/978-3-030-31321-0_15.

[24] A. Korichi, S. Slatnia, N. Tagougui, R. Zouari, M. Kherallah, and O. Aiadi, "Recognizing Arabic Handwritten Literal Amount Using Convolutional Neural Networks," *Lect. Notes Networks Syst.*, vol. 413 LNNS, pp. 153–165, 2022, doi: 10.1007/978-3-030-96311-8_15.

[25] S. Al-Ma'adeed, D. Elliman, and C. A. Higgins, "A data base for Arabic handwritten text recognition research," *Proc. - Int. Work. Front. Handwrit. Recognition, IWFHR*, vol. 1, no. 1, pp. 485–489, 2002, doi: 10.1109/IWFHR.2002.1030957.

[26] A. Abdu, Z. Zhai, R. Algabri, H. A. Abdo, K. Hamad, and M. A. Al-antari, "Deep Learning-Based Software Defect Prediction via Semantic Key Features of Source Code—Systematic Survey," *Mathematics*, vol. 10, no. 17, 2022, doi: 10.3390/math10173120.

[27] N. Aouadi and A. K. Echi, "Word Extraction and Recognition in Arabic Handwritten Text," vol. 12, no. 1, pp. 17–23, 2016.

[28] Q. Al-Nuzaili, A. Hamdi, S. Z. M. Hashim, F. Saeed, and M. S. Khalil, "An enhanced quadratic angular feature extraction model for arabic handwritten literal amount recognition," *Lect. Notes Data Eng. Commun. Technol.*, vol. 5, pp. 369–377, 2018, doi: 10.1007/978-3-319-59427-9_40.

[29] N. Aouadi and A. K. Echi, "Word Extraction and Recognition in Arabic Handwritten Text," *Int. J. Comput. Inf. Sci.*, vol. 12, no. 1, pp. 17–23, 2016, doi: 10.21700/ijcis.2016.103.