Multi-CNN Voting Method for Improved Arabic Handwritten Digits Classification

Abstract—**Handwritten Arabic recognition poses a unique set of challenges due to the intricate nature of the script, its diverse styles, and the inherent complexity of the Arabic language. With the increasing need for automated recognition systems in various domains, such as document analysis, postal services, and historical document preservation, finding the best system for accurate and efficient recognition becomes crucial. The primary objective of this paper is to propose a method for enhancing the performance of handwritten Arabic written classification tasks by incorporating ensemble learning by employing multiple CNN classical network features, including Inception, Lenet-5, AlexNet, Dense, MobileV2, and ResNet, and selecting the final results through a voting approach. To validate the effectiveness of our proposed method, we evaluated it on the MADBase dataset and compared it with the latest works in the field. The results demonstrate that our proposed method achieved better accuracy, F1-score, precision, and recall, with scores of 99.31, 99.31, 99.31, and 99.30, respectively, outperforming existing research. This proposed method can be adapted and utilized for various classification tasks. [1].**

Index Terms—**Classical CNN, MADBase, ResNet, ensemble learning, convolution neural networks, handwritten digits detection**

I. Introduction

The Arabic language is widely regarded as one of the most popular languages in the world, with a staggering 300 million speakers, making it the 4th largest spoken language globally [1]. It has spread beyond its origins in the Middle East and North Africa to other continents as well. Most of these countries use Arabic numerals for identification cards, car numbers, street signs, and various other important documents [2].

The use of artificial intelligence (AI) has revolutionized various industries and how we live our lives [3], [4]. One of the most significant applications of AI is automatic digit recognition, which has become increasingly vital across different fields [5]. Automated numerical recognition is highly advantageous over human recognition, as it can quickly process vast amounts of data. Additionally, it requires less human effort, making it ideal for tasks like car number tracking and other activities that traditionally require significant manual labor. Despite being faster than human recognition, automated numerical recognition systems are not immune to errors. Even a small mistake can be incredibly costly in certain applications, such as identification cards at ports and other important documents. Therefore, it is crucial to use the most advanced and intelligent systems with high accuracy to recognize numbers. While there are many studies focused on increasing performance across various number languages, there have been relatively few studies dedicated to improving the recognition of Arabic numbers. In [6] the authors utilised a variety of convolution neural network algorithms, including AlexNet, ResNet-18, and GoogleNet, to extract relevant features from Arabic images. These features were then passed to a machine learning classifier to categorise the images. The researchers employed three different machine learning techniques, namely support vector machines (SVM), K-nearest neighbours (KNN), and decision trees (DT), to further improve the classification accuracy. The MADBase dataset was used for the experiment, and they achieved 96 accuracy. In [7] the sliding window technique was used to slide over an image, and at each position of the window, a block was extracted from the image. Then, the features of each block were extracted using one of the proposed feature extraction techniques, including mean-based, gray-level co-occurrence matrix (GLCM), moment-based, and edge direction histogram (EDH). After feature extraction, the features of all the blocks were combined, and

[1] The link will be available after the anonymity period

the resulting feature vector was used as input to the machine learning classifiers (random forests and support vector machines) for digit recognition. The author used the MADBase dataset and achieved 98.33 by using mean-based and moment-based methods with using RF, while they achieved 99.13 by using EDH and GLCM with SVM. In [8], the author ututilized CNN architecture with seven layers, comprising three convolutional layers and three max-pooling layers positioned between each convolutional layer, followed by a fully connected layer with softmax activation. The intention was to train and assess the model's performance using the MADBase dataset, and to compare the results with those from other research papers. The CNN architecture implemented by the author achieved an accuracy of 98.95. In [9], the author proposed a hybrid transfer model consisting of two pre-trained convolutional neural networks (CNN) and recurrent neural networks (RNN) with long-short-term memory (LSTM) architectures. The CNN models learned the relevant features of Arabic (Indian) digits, and the LSTM layers extracted long-term dependence features. The proposed model was trained and tested using MADBase and achieved an accuracy of 98.92.

In an effort to enhance and create a highly accurate system for classifying Arabic digit handwriting, researchers have developed numerous strategies. Nevertheless, despite these efforts, a system with consistently good classification performance has not yet been developed. In order to overcome this difficulty, we suggest and test the viability of a voting-based ensemble learning strategy in this study. Our research shows that this strategy significantly improves classification performance, leading to better accuracy rates.

This paper is organized as follows. Section II explains the dataset used in this study, while Section III provides a brief introduction to the convolutional neural network (CNN) and the network architectures used. Section IV covers the design and implementation of the proposed method architecture. In Section V, the performance of the proposed method is compared with other network architectures. Finally, the last section presents the conclusion of the paper.

II. MADBase Dataset

The quality and quantity of the dataset used for training and testing are critical factors that determine the effectiveness of a deep learning system. A diverse and representative dataset of high quality can help a system learn underlying patterns and generalise effectively, resulting in better performance. In this paper, we used the widely popular MADBase dataset of handwritten Arabic digits, which was collected and prepared by [10] and has been extensively used in prior related work. The dataset comprises four files, containing a total of 70,000 samples, of which 60,000 are for training and 10,000 are for testing. Each sample is a PNG image with a size of 28 x 28 pixels. Figure 1 shows a representative slice of the dataset.

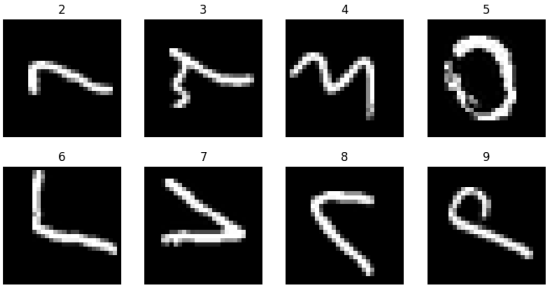

Fig. 1: Arabic Digits Samples from MADBase Dataset

III. Classical networks

In this paper, we utilised five of the most popular classical convolution network architectures: AlexNet, MobileNetV2, ResNet, Inception, and LeNet-5. Each of these CNNs has been modified and adapted to be suitable for the input size of the dataset, which is 28x28 pixels. Below is a brief description of each of these networks and their key characteristics.

1) LeNet-5: Developed by Yann LeCun and his colleagues in the 1990s, this was one of the first convolutional networks, and was designed for handwritten digit recognition. It was a breakthrough in computer vision at the time, and set the stage for future developments in deep learning [11].

2) AlexNet: Developed by Alex Krizhevsky and his colleagues in 2012, this was one of the first deep convolutional networks, and won the ImageNet competition that year. Its success helped to spark the current deep learning revolution, and demonstrated the power of convolutional networks for image recognition [12].
3) Inception: Developed by Google, this architecture is characterised by its use of ”inception modules”, which use multiple convolutional filters at different scales to capture rich features. This approach allows for efficient use of computational resources [13].
4) ResNet: Developed by Microsoft, this architecture introduced the concept of residual connections, which allow gradients to flow more easily during training, and enable very deep networks to be trained effectively. ResNet has achieved state-of-the-art results on many benchmarks [14].
5) MobileNetV2: Developed by Google researchers in 2018, this architecture is designed to be lightweight and efficient, making it ideal for mobile and embedded devices. It uses depthwise separable convolutions to reduce computation and memory requirements while maintaining accuracy, and has been widely adopted in mobile applications [15].

In addition to these architectures, we also evaluated the dataset by using a dense deep learning network

IV. Proposed method

This section consists of two parts: the first part presents our proposed method idea, and the second part outlines the characteristics that define our proposed method.

A. System overview

The proposed method is inspired by the random forest mechanism [16]. A random forest consists of multiple decision trees. During training, the data is split into multiple subsets, and each subset is fed to a decision tree for training. During testing, the dataset samples are fed to all the decision trees, and each tree generates an individual result. The most common result is then selected as the final predicted value. In this method, we use this method and consider each one of the classical convolution networks as an individual tree. We train each convolutional network on the dataset. Once the CNNs are trained, during testing, the sample enters all the CNNs, and similar to the decision tree, we select the most common value as the final predicted value.

In some cases within the proposed method, there is a possibility that each of the three algorithms may generate a common prediction. For example, the LeNet-5, AlexNet, and Inception algorithms might predict a value different from the results of the other three networks (Inception, Resnet, and Dense). Alternatively, each of the two networks could predict a value that differs from the other two networks. In such instances, we have decided to select the predicted value of the ResNet network as it has demonstrated the highest accuracy when trained individually, as highlighted in Section IV. Figure 2 displays the proposed system flowchart.

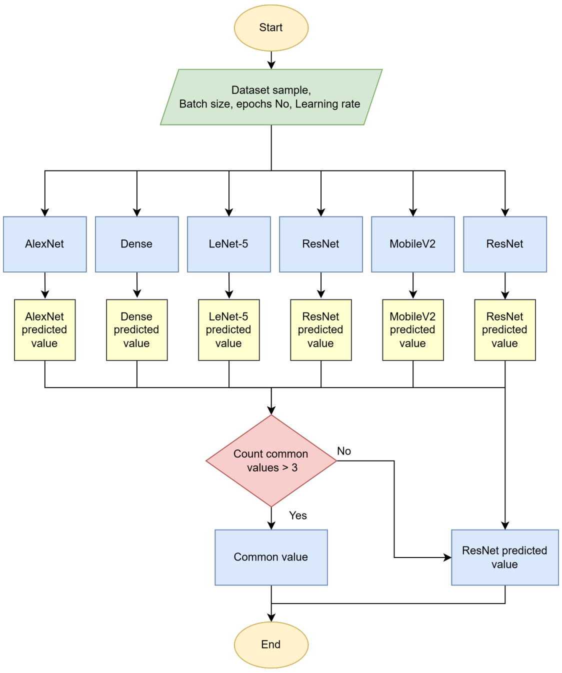

Fig. 2: Proposed method flowchart

B. Proposed method significance

In the following the main characteristic of using the proposed method

1) Each CNN architecture has its own unique characteristics. For example, ResNet is often used for regularisation to prevent overfitting, while Inception is designed to detect deeper features. By using a voting method with multiple CNN architectures, the proposed method can leverage the strengths of each network and extract more meaningful information from the dataset resulting in higher accuracy.
2) The proposed method is not limited to Arabic handwritten classification and can be adapted to work for any classification task.
3) The proposed method is not limited to CNN networks; it can be used with various machine learning techniques, including RNN and others. For example, the system can utilise multiple machine learning techniques to classify a specific task. In this way, as mentioned in point 1, we can take advantage of all machine learning techniques features since all of them will be included in the make the decisions.

V. Performance evaluation

A. Performance metrics

In this section, we conducted experiments with various hyperparameters to fine-tune the classical networks. Each CNN network has trained with two epochs (10 and 20) and three different learning rates (0.0001, 0.0005, and 0.001), along with different batch sizes (64, 128, 192) for each learning rate. We selected the hyperparameters for each network that achieved the highest performance. We constructed the proposed method using the CNN networks configured with these high-performance hyperparameters. Table I presents the highest accuracy, F1-score, recall, and precision achieved by each CNN network using specific , which were compared with the proposed method. Results indicate that combining multiple CNN networks and selecting the common predicted value achieved superior performance compared to using individual CNN networks, as demonstrated by all performance metrics.

B. Time consumption

We evaluated the time required to execute one sample of the dataset for each method and found that the proposed method requires more time compared to other methods. However, all methods, including the proposed method, were able to complete the task in less than 0.32 seconds, which is acceptable. Additionally, with the advancement of hardware technology, the execution time can potentially be further reduced, and any impact on the system's overall performance is expected to be negligible. Figure 3 shows the time consumption required for each method to complete one task.

Table II presents a comprehensive comparison between the proposed method, which employs ensemble learning with multiple CNN classical network features, and recent works in the field of handwritten Arabic classification. The proposed method shows high performance compared with recent works.

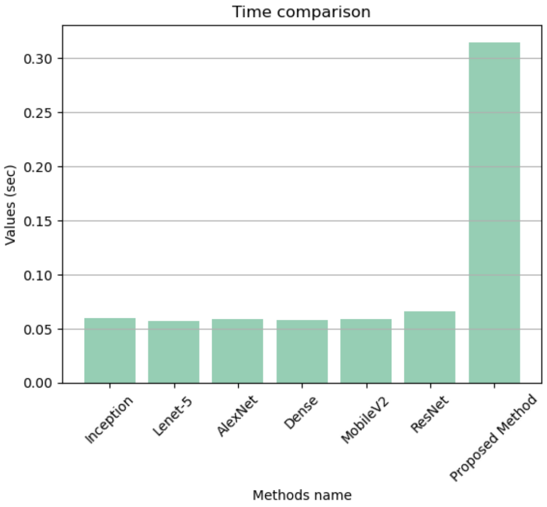

Fig. 3: Performance Evaluation of CNN Networks based on Execution Time for Single Task

VI. Conclusion

In conclusion, we have proposed a method for handwritten Arabic digit recognition based on training multiple CNN architectures and using a voting mechanism for prediction. Our proposed method outperforms classical models and related work models in terms of accuracy, F1-score, recall, and precision. While our proposed method

TABLE I: Comparing the performance of our proposed method with CNN networks.

Methods	LR	Epochs	Batch size	Acc%	F1 %	Prec %	Rec%
Inception	0.0005	10	128	98.92	98.93	98.93	98.94
LeNet-5	0.001	20	128	98.79	98.80	98.80	98.79
AlexNet	0.0001	20	128	98.81	98.82	98.82	98.81
Dense	0.0005	20	192	98.09	98.09	98.11	98.09
MobileV2	0.001	10	192	98.54	98.55	98.58	98.54
ResNet	0.0001	10	192	99.02	99.02	99.02	99.02
Proposed method				99.31	99.31	99.31	99.30

TABLE II: Comparison of Methods for Handwritten Arabic digits classification

Method	Authors	Dataset	Accuracy (%)
CNN and machine learning	Bashar and et al [6]	MADBase [10]	96.00
Sliding window and GLCM	Ebrahim and et al [7]	MADBase [10]	99.14
CNN and LSTM	Rami and et al [9]	MADBase [10]	98.92
Voting based	proposed method	MADBase [10]	99.31

may consume slightly more time compared to other CNN networks, the time delay is minimal, taking less than 0.4 seconds. The proposed method can be applied in systems where accuracy, precision, F1-score, and recall are high priorities. The proposed methods can be adapted to work with different methods.

References

[1] Abdelaziz A Abdelhamid, Hamzah A Alsayadi, Islam Hegazy, and Zaki T Fayed. ”end-to-end arabic speech recognition: A review”. In *Proceedings of the 19th Conference of Language Engineering (ESOLEC'19), Alexandria, Egypt*, pages 26–30, 2020.

[2] Yasir Elhadi, Omar Abdalshakour, and Sharief Babiker. Arabic-numbers recognition system for car plates. In *2019 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, pages 1–6. IEEE, 2019.

[3] Bernd Carsten Stahl. *Artificial intelligence for a better future: an ecosystem perspective on the ethics of AI and emerging digital technologies*. Springer Nature, 2021.

[4] Francisco J Cantú-Ortiz, Nathalíe Galeano Sánchez, Leonardo Garrido, Hugo Terashima-Marin, and Ramón F Brena. An artificial intelligence educational strategy for the digital transformation. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 14:1195–1209, 2020.

[5] Jie Zhang, Shu Liang Li, and Xi Liu Zhou. Application and analysis of image recognition technology based on artificial intelligence–machine learning algorithm as an example. In *2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL)*, pages 173–176. IEEE, 2020.

[6] Bashar Al-Saffar, Amjed R Al-Abbas, and Selma Ayşe Özel. A comparative study on the recognition of english and arabic handwritten digits based on the combination of transfer learning and classifier. In *Proceedings of the 2nd International Conference on Emerging Technologies and Intelligent Systems: ICETIS 2022, Volume 2*, pages 95–107. Springer, 2022.

[7] Ebrahim Al-wajih and Rozaida Ghazali. Improving the accuracy for offline arabic digit recognition using sliding window approach. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, 44:1633–1644, 2020.

[8] Jawad H Alkhateeb. Handwritten arabic digit recognition using convolutional neural network. *International Journal of Communication Networks and Information Security*, 12(3):411–416, 2020.

[9] Rami S Alkhawaldeh. Arabic (indian) digit handwritten recognition using recurrent transfer deep architecture. *Soft Computing*, 25(4):3131–3141, 2021.

[10] Ahmed El-Sawy, Hazem El-Bakry, and Mohamed Loey. Cnn for handwritten arabic digits recognition based on lenet-5. In *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2016 2*, pages 566–575. Springer, 2017.

[11] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[13] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings*

of the IEEE conference on computer vision and pattern recognition, pages 4510–4520, 2018.

[16] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.