# Benchmarking Lightweight Convolution Neural Networks for Children's Arabic Handwriting

**Khalil Ladrham[1], Hicham Gueddah[2], Brahim Ouben Hssain[3]**

[1]Intelligent Processing and Security of Systems, Faculty of Sciences, Mohammed V University, Rabat, Morocco.
[2]Intelligent Processing and Security of Systems Team, E.N.S, Mohammed V University, Rabat, Morocco.
[3]Laboratory of Science and Technology for the Engineer, LaSTI-ENSA, Sultan Moulay Slimane University, Khouribga, Morocco.

**E-mail:** [1]khalil_ladrham@um5.ac.ma, [2]h.gueddah@um5r.ac.ma, [3]OUBENHSSAIN.Brahim@usms.ac.ma

## Abstract

The children cannot recognize the Arabic script handwriting because the recognizer is intolerant of high inter-individual variations. The issue is further compounded by other problems such as irregular strokes, interrupted forms and inconsistent aspect ratios. The study explores which CNN architectures are most suitable for robustly recognizing black-and-white Arabic letters and digits drawn from kids, aged 5-10. The dataset comprises 570 000 characters images. The six well known CNN architectures are LeNet 5, AlexNet VGG16, GoogLeNet, DenseNet, and ResNet50. In order for the experiments to be reproducible and easily verifiable, we used a supercomputer for all models training and tests. The ResNet50 model was shown to perform best of all models with a validation accuracy of 99.86%, a global F1 score of 99.89%, validation loss of 6% and 0.96 GFLOPS. Along with benchmarking, the proposed work provides optimized lightweight CNNs for 64×64 grayscale images of children's handwritten Arabic characters. The suggested model achieves a recognition accuracy of 98.3% at a cost 41% lower than VGG16, while drastically reducing the number of parameters. According to the study, modeling various handwritten text types can benefit from residual learning. LeNet-5 and other lightweight models have demonstrated good performance with less processing power and can be applied to embedded systems. The results indicate that children's handwriting can be automatically analyzed using the improved CNNs. Additionally, they demonstrate the applicability of these CNNs in digital assessment systems, educational technologies, and multilingual writing processing. The study offers an AI method for interpretability and robustness and lays the foundation for hybrid CNN-Transformer models.

**Keywords:** Arabic Characters, Handwritten, Convolution Neural Networks, Optical Character Recognition, CNN-Transformer, Interpretability.

## 1. Introduction

In Morocco, as throughout North Africa, Arabic is widely used in writing, with more than 37 million people using Arabic script. However, numerical notation mainly uses numerals in education, public administration and the media [1].

*DOI: https://doi.org/10.36548/jiip.2026.1.012*
*Received: 18.12.2025, received in revised form: 16.01.2026, accepted: 02.02.2026, published: 11.02.2026*

Handwritten Character Recognition (HCR) represents one of the most important and attractive areas of Natural Language Processing (NLP). To ensure the provision of several essential services, such as document digitization, banking transaction management, postcode translation, and the memorization of various types of handwritten documents, this domain has been the subject of much research [2], [3], [4].

Handwritten Arabic poses significant challenges for recognition due to its continuous nature and variable letter shapes. Letters depend on different contexts, such as when they are in the middle, at the end, at the beginning or alone. Distinguishing Arabic numerals is challenging due to the influence of punctuation and diacritical marks on their meanings, the similarity of numerous characters, and the variation in numeral representation based on writing style [5]. In order to automate tasks such as sorting mail and number plates, one advantage of this approach is its ability to train effectively on large-scale datasets. However, this can be challenging in places where people speak more than one language, as different handwriting styles and scripts can make them difficult to understand [6], [7], [8].

Children's handwriting is harder to read because it contains more broken lines, uneven proportions, reversed character shapes, and differences among classes. Models trained on adult handwriting data show limited generalization to children's handwriting examples. Therefore, it is important to develop automatic recognition of handwritten letters [9], [10].

Accurate recognition of children's handwritten Arabic letters directly affects the effectiveness of learning techniques, such as automatic handwriting exams, early identification of learning problems like dysgraphia, and providing feedback to young children. Accurate and reliable handwriting recognition systems may help digitize many instructional materials, automate tests, and create entire online learning systems in multiple languages with minimal resources.

Asynchronous handwriting recognition still relies on CNN model. The families that LeNet-5, AlexNet, VGGNet, GoogLeNet (also called Inception) ResNet, and DenseNet, represent are feature extractors often balanced in accuracy and prediction costs. Numerous studies report latency and accuracy issues that influence the choice of architecture [11].

During the feature extraction process, the principal hurdles of various systems can be recognized. Most systems suffer from long training times and the configuration of CNN parameters [12]. The main objective of this study is to examine the handwriting of young children’s characters. The performance of modern CNNs (see Table 1) was evaluated on data of Arabic characters collected from children aged 5 to 10 years.

The work ranges from Arabic consonants to mixed scripts with Arabic consonants written by children and expands on previous work presented at IA&NLP 2024, the third international workshop on Artificial Intelligence for Natural Language Processing held in Leuven, Belgium [13].

The study evaluates the robustness of CNNs from various representative families by building a child-specific Bangla handwritten character corpus. It uses LeNet-5, AlexNet, VGGNet16, ResNet50, DenseNet, and GoogLeNet. This is carried out through the use of a standard protocol for pre-treatment and grayscale normalization, equilibrated splits with augmentation, and confusion matrix analysis [14].

**Table 1.** Arabic Characters in Latin Transliteration

| Arabic Characters | Latin Transliteration | Class Number | Arabic Characters | Latin Transliteration | Class Number | Arabic Characters | Latin Transliteration | Class Number |
|---|---|---|---|---|---|---|---|---|
| 0 | Sifr | 0 | ث | Tha | 13 | ظ | Za. Emph | 26 |
| 1 | Wahid | 1 | ج | Djim | 14 | ع | Ain' | 27 |
| 2 | Ethnan | 2 | ح | Ha | 15 | غ | Ghayn | 28 |
| 3 | Thalatha | 3 | خ | Kha | 16 | ف | Fa | 29 |
| 4 | Arba'a | 4 | د | Dal | 17 | ق | Qaf | 30 |
| 5 | Khamsa | 5 | ذ | Dhal | 18 | ك | Kaf | 31 |
| 6 | Sitta | 6 | ر | Ra | 19 | ل | Lam | 32 |
| 7 | Sab'a | 7 | ز | Taï | 20 | م | Mim | 33 |
| 8 | Thamania | 8 | س | Sin | 21 | ن | Noun | 34 |
| 9 | Tis'a | 9 | ش | Shin | 22 | هـ | Ha | 35 |
| أ | Alif | 10 | ص | Sad | 23 | و | Waw | 36 |
| ب | Ba | 11 | ض | Dad | 24 | ي | Ya | 37 |
| ت | Ta | 12 | ط | Ta Emph | | | | |

To find a compromise between speed and accuracy, several factors go into choosing the best recognition technique for analyzing handwritten characters. These include the size of the dataset, the diversity of children's handwriting within a class, the cost of computing, and the necessity for real-time implementation of lightweight CNN architectures.

This study has used a similar approach to that of Alsayed [14]. The HCR is performed in eight steps: (I) Data Acquisition, (II) Pre-processing, (III) Data Augmentation, (IV) Training Model, (V) Feature Extraction, (VI) Classification Layer, (VII) Evaluation and (VIII) Comparative Analysis. Figure 1 illustrates the workflow of the CNN based recognition of Child Handwriting.

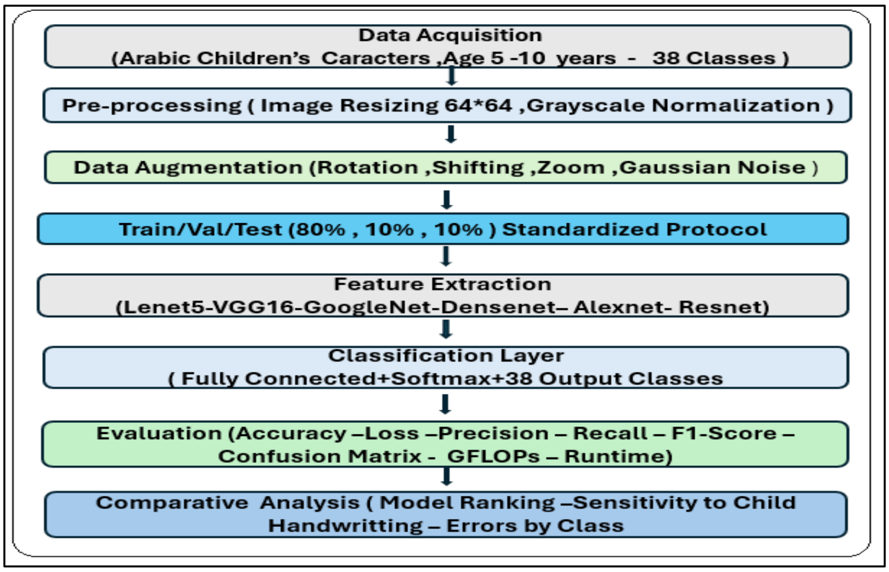

Data Acquisition
(Arabic Children's Caracters ,Age 5 -10 years - 38 Classes )
Pre-processing ( Image Resizing 64*64 ,Grayscale Normalization )
Data Augmentation (Rotation ,Shifting ,Zoom ,Gaussian Noise )
Train/Val/Test (80% , 10% , 10% ) Standardized Protocol
Feature Extraction
(Lenet5-VGG16-GoogleNet-Densenet– Alexnet- Resnet)
Classification Layer
( Fully Connected+Softmax+38 Output Classes
Evaluation (Accuracy –Loss –Precision – Recall – F1-Score –
Confusion Matrix - GFLOPs – Runtime)
Comparative Analysis ( Model Ranking –Sensitivity to Child
Handwritting – Errors by Class

**Figure 1.** The Workflow for Recognizing Children's Handwriting Based on CNNs by Authors

This paper focuses on CNNs, which deliver an optimal combination of prediction cost,accuracy, and computational power in CPU-only settings such as educational platforms and low-power appliances. The contributions of this paper are: (1) constructing and testing one of the largest handwriting datasets for Arabic children; (2) comparing six families of CNNs in terms of accuracy, loss, precision, recall, F1 score, execution time, and FLOP; (3) identifying

ResNet50 as the most precise and efficient architecture for this problem; (4) proposals for future hybrid methods based on CNN-Transformer and explainable AI to enhance understandability.

It introduces a comprehensive experimental methodology that focuses on the basic issues involved in the recognition of pupils' handwriting. The main contributions of this work are:

- A novel evaluation of convolutional neural networks on a dataset of 570,000 Arabic characters handwritten by children aged 5 to 10 years, with statistical and reproducible results.
- A comparison of shallow and deep convolutional neural network architectures, showing their advantages and disadvantages in terms of cost, accuracy, and model complexity.
- Systematic integration of computational complexity metrics (e.g., FLOPs, time per epoch) to allow for a fair and sensitive comparison of model hardware.
- The experimental results validate the effectiveness of lightweight convolutional neural networks in limited-resource learning contexts.

## 2. Related Work

Several CNN-based handwriting recognition systems have been suggested in the literature. Michalski and Plechawska-Wójcik tested LeNet-5, AlexNet, and GoogLeNet on MNIST digits and EMNIST letters. When evaluated on single-digit test datasets, GoogLeNet attained the highest accuracy of 99.04% on MNIST and 94.31% on EMNIST, while LeNet-5 achieved the highest inference time of 1.3 s for 10k MNIST digits and 1.9 s on EMNIST, suggesting a trade-off between runtime and accuracy [11].

Ali and Abdulrazak compared AHDBase (70,000 Arabic digits) and a custom CNN on KurdSet with DenseNet121, ResNet50, and MobileNet. They set an accuracy efficiency benchmark for digit OCR on the datasets with their reported test accuracy of 99.73% on KurdSet (DenseNet121 and the custom CNN) and 99.74% on AHDBase (ResNet50), with an almost perfect F1/AUC after preprocessing [15].

Finjan et al. presented a system for recognizing Arabic handwritten digits based on transfer learning with a pre-trained ResNet-34, tested on MADBase (60,000 training images and 1,000 test images). Their pipeline uses grayscale and background digit inversion and trains quickly with a single-cycle strategy. The model obtains a test accuracy of 99.6%, outperforming the MLP/CNN baselines (93.8-99.4%) and demonstrating the efficacy of residual networks for Arabic optical character recognition [16].

Al-Maamari et al. proposed a classification of Arabic characters by combining VGG16 and ResNet50 with an encoder transformer using Arabic Char-4k Plus AHCR; it achieves 99.51% and 98.19%, outperforming single-model CNNs and a benchmark set. An approach trained on Arabic Char-4k and tested on AHCR yields an accuracy of 75.40%, which implies significant global generalizations [17].

Alwagdani and Jaha explored methods for teaching children (Hijja) and adults (AHCD). They developed a CNN that differentiates between children's handwritten Arabic letters.

Training on Hijja achieved an accuracy of 92.96% (SVM/SoftMax), higher than that of children alone (91.95%) and adults alone (80.17%). They employed HOG and statistical indicators to detect the difference between youngsters' and adults' handwriting to improve the CNN's performance. This raised the overall percentage to approximately 90–94%. This highlights how vital it is to mix features and training approaches [9].

Hussain et al. employ spiking neural networks (SNNs) and convolutional SNNs (CSNNs, soft-LIF/STDP) to build biologically feasible Arabic handwritten digit recognizers, with an emphasis on both neuromorphic compatibility and energy efficient processing. By comparing accuracy-performance trade-offs with classification time, synaptic constants and spike rates, the CSNN performs at 98.98% accuracy and the STDP-SNN at 91.16% on ADBase (70k images, 32×32), approaching non-spiking benchmarks (e.g., VGG 99.57%) [18].

De Sousa et al introduce a VGG-based framework for offline Arabic handwriting. MADBase (numbers) and AHCD (letters) perform at 99.74% and 98.42% accuracy, respectively. They observe that VGG families achieve the best performance on classical datasets [6].

Arif and Poruran train the AlexNet algorithm using Transfer Learning (TL), and evaluate it on the IFHCDB. TL-AlexNet scores 96.3% on IFHCDB and OCR-GoogLeNet scores 94.7%; training performance on comparable datasets is close to maximal performance. The results make AlexNet a reference point for Arabic handwriting TL [7].

Plechawska-Wójcik tests LeNet-5, AlexNet and GoogLeNet on the MNIST and EMNIST databases. GoogLeNet is the most accurate 99.04% on MNIST and 94.31% on EMNIST, but LeNet-5 is the fastest at 1.3 s for 10k MNIST and 1.9 s for EMNIST. The study showed a clear trade-off between accuracy and runtime across CNNs [11]. These studies have limitations discussed in (see Table 2).

**Table 2.** Limitations of Recent Studies from 2020-2025

| Study | Limitations |
|---|---|
| [6] | Including gaps or segmentation in MADBase/ADBase datasets, reliance on individual studies using a single dataset, non-tested layout/segmentation, and poor generalization within datasets. |
| [9] | Training on Hijja and AHCD revealed the best performance on Hijja (92.96%; SVM/SoftMax); testing on children only 80%. 32×32 Low resolution, evaluation limited to these two datasets, generalization not tested, separated letters only |
| [11] | The evaluation was executed on a unique computational platform (Windows 10 Pro, i7-6700, 16 GB RAM) and only includes 99.04% accuracy on MNIST and 94.31% accuracy on EMNIST Letters. CPU Tests Time, and does not estimate overfitting. |
| [15] | Only digital corpus for AHDBase data with 99.74% accuracy, evaluation limited to a single split of two datasets, and no evaluation on external generalization. |
| [16] | A ResNet-34 model transfer learning reaches 99.6% accuracy on the test set when trained on 60,000 handwritten Arabic numbers and tested on 1,000. The study is constrained to a single-digit dataset and offers accuracy metrics without considering loss rates, overfitting, or generalization analysis. |
| [17] | The Char-4k Arabic database is 75.40% accurate, yet there were times when characters that appeared the same were mixed up. But the studies do not use greater sets of data |
| [18] | ADBase thinks that the CSNN scores will be 98.98%. There are 70,000 handwritten numerals on a grid that is 32x32.They use 60,000 for training and 10,000 for testing. Nonetheless, the literature fails to include larger datasets. The study is limited to a 32x32 dataset containing a single digit and does not assess the dataset's suitability for cross-validation. How to achieve it. |

## 3. Methodology

### 3.1 Data Preparation and Partitioning

This data consisted of 570,000 black-and-white images of handwritten Arabic letters (28 letters and 10 digits, 38 classes). The data used in this paper was collected as part of the original data collection procedure in the primary schools of the Essaouira area and the Safi region of Morocco. Samples comprising the handwriting of children aged between 5 and 10 years were collected in collaboration with school staff, only after obtaining informed consent from their parents. All of the samples were blinded and no information that could be traced to the individual was stored at the time of collection. As far as they could understand, this dataset has not been reported in the past and has been collected solely for this research. The pictures were resized to 64 x 64 pixels and normalized to the range of -1 to 1. To ensure the stability of the results, translation (±5%), zoom (±5%), minor rotations (±7 degrees), and light Gaussian noise were applied to balance the classes and to avoid overfitting. The data was separated into three parts: 80% training, 10% validation, and 10% testing. The classes were chosen randomly with set random seeds. Fig. 2 below contains some examples of the letters and numbers from the handwriting dataset of the children learning the Arabic language. The choice of 64 x 64 as the resolution of the scanned images was based on the trade-off between the cost and benefits of maintaining the properties of discriminative strokes versus lower computing costs. One advantage is the ability to be trained effectively on large datasets. Data enhancement may include small cycles and simple noises to ensure that the content of the Arabic letters is not further degraded. Nevertheless, the model demonstrates resistance to large intra-class variance due to the extensive range of the dataset and the ability of CNNs to extract features at various levels. With batch normalization and dropout, models can learn to represent many different characters, even when the handwriting of children is abnormal, as these models are able to generalize.

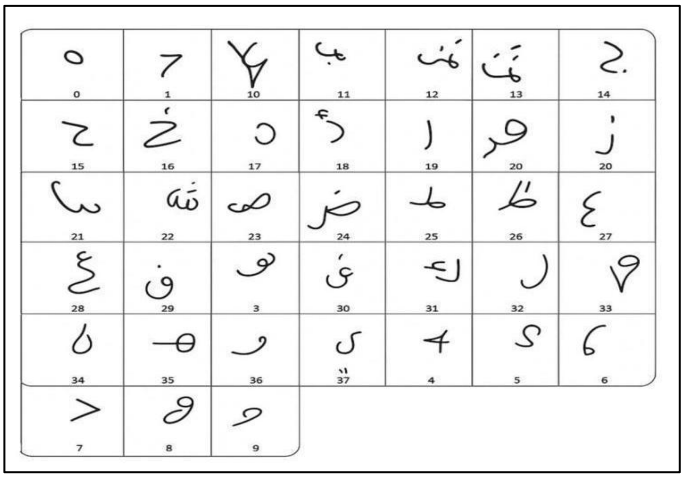


**Figure 2.** Some Samples from the Used Dataset

### 3.2 Model Architectures

CNNs are among the most powerful deep learning algorithms capable of handling large data sets. Their hierarchical structure automatically filters features from the lower levels to the higher levels. CNNs can manage ImageNet, which consists of over a million images, by creating classifiers that can differentiate between categories. Through the employment of convolution filters, pooling, and non-linear activations, CNNs can deal well with disordered handwriting and real-world images. Due to their capability of both sharing parameters and reaching into hundreds of layers on contemporary hardware, CNNs are more cost-effective to train than fully connected networks [19]-[20].

CNNs are employed in this study because the identification of the intrinsic pattern of space is inherent with HCR. They enable hierarchical feature acquisition starting with low level stroke and edge detection and stepping  up to high-level structure representations, retaining the two-dimensional topology of picture data. Handwriting, due to its unstable proportions, shaky lines, and high intra-class variance, is particularly in need of this quality, among children. The architectural enhancements investigated in this work are the result of the selection of representative CNN families that utilize different optimization processes. Residual learning in ResNet50 has identity shortcut connections, and DenseNet superimposes restrictions on parameter development with the inclusion of feature reuse, linking each layer to all subsequent layers in ResNet50 to help with gradient propagation and prevent performance degradation in deep networks. Conversely, GoogLeNet's elementary modules and DenseNet recommend extracting multi-scale features through Inception modules [21]. Additionally, the complexity trade-off, computational cost and recognition accuracy trade-off based on the use of LeNet-5, VGG16, and AlexNet as a baseline design can be assessed. The best convergence is guaranteed by optimization under AdamW, and additional elements such as batch normalization and dropout optimisation are used to enhance all designs by stabilizing the training and decreasing overfitting. This set of design choices for a comparison of CNN architectural designs through the same experimental method in a significant and fair manner.

### 3.3 Training Protocol

All tests are done on a CPU, which is spread across eight units in the Marwan Cluster. All the models are trained using the AdamW optimizer, which had an initial learning rate of 1e−3 and a weight decay of 5e−3.The amount of CPU RAM available was dependent on the architecture of the device and the quantity thereof. Early termination and a learning rate schedule are used to ensure that convergence remains constant.

The batch sizes in Table 3 are selected to find a balance between the depth of the model and the CPU memory and training stability of the high-performance computing (HPC) cluster. LeNet-5 has a few parameters and a small memory footprint, enabling it to run larger batch sizes (e.g. 256) which makes it more stable in convergence on deeper network as well as being efficiently used in a CPU-only setting, providing a consistent and fair comparison between architectures.

**Table 3.** Batch Size Used in Each CNN Architecture

| Model | LeNet-5 | VGG 16 | GoogleNet | DenseNet | AlexNet | ResNet 50 |
|---|---|---|---|---|---|---|
| Batch size | 256 | 64 | 64 | 32 | 128 | 32 |

The training/validation accuracy, loss, macro/micro F1 scores and confusion matrices were recorded for each run. The study compared the models used under the conditions of accuracy, precision, recall, and the F1-score [22]. Micro-averaged and macro-averaged measures are likely to yield similar outcomes when the distribution of the classes is significantly equal and when there is consistent per-class performance. This is due to the fact that both measures will depict standardized categorization behaviour between classes. To calculate the performance indicators such as runtime per image and FLOPs, the classification metrics are employed to determine the efficiency and complexity of each design. Execution time per image is the mean time taken to classify one sample; it is applicable in real-time. The FLOPs metric is used to indicate the count of floating-point operations in a single forward pass, reflecting the costliness and scalability of the network. These complementary actions help to provide an equitable evaluation of the trade-off between model accuracy and computational efficiency. These measurements and performance indicators are portrayed in the equations (1), (2), (3), (4), (5) and (6).

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

The recognition error rate is included in this study, which is a standard way to measure how often characters are misclassified in optical character recognition, $Error = 1 - Accuracy$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1 - Score = \frac{2*Precision*Recall}{Precision+Recall} \quad (4)$$

$$FLOPs = \sum_{l=1}^{L} 2 * C_{in}^{l} * C_{out}^{l} * (K^{l})^{2} * H_{out}^{l} * W_{out}^{l} \quad (5)$$

$$Runtime_{epoch} = \frac{N_{train}*FLOPs_{model}}{B*R_{CPU}*N_{nodes}} \quad (6)$$

The computational cost of training is given as the average time per epoch to estimate the runtime and enable efficient comparison of the performance of different designs. A better measure of inference running time per picture is useful in real-time deployment applications because it directly indicates the amount of time it will take to make a prediction in practice.

The data was separated into three sections: testing, validation, and training. The weights, addition of new data, and the mix of data are updated with the help of a randomly selected seed of 42. Padding was included to maintain the aspect ratio and ensure the size of the pictures was 64x64 pixels; they are shrunk to fit within the range [0, 1]. An example would be to add random rotation of up to ±7 degrees, horizontal and vertical motion of up to ±5%, a zoom variation of up to ±5%, and low-contrast Gaussian noise. All models are trained using the AdamW optimizer. The weight decay is 5x10$^{-3}$, and the initial learning rate is 1x10$^{-3}$. The batch size may range from 32 to 256, depending on the complexity of the model and the type of hardware it can be applied to. When the testing loss is low, training may end before the 50th epoch takes place. It is trained with the help of a ReduceLROnPlateau scheduler (patience is three epochs, and the reduction factor is 0.5). To demonstrate the results using both micro- and macro-averaging techniques, F1-score, recall, accuracy, and precision are employed. The performance of each group and the frequency of incorrect answers given by participants in response to the questions can be viewed with the help of confusion matrices (Figure 3). A detailed plan of the proposed convolutional neural network (CNN)-based system for identifying handwritten Arabic letters is provided in Figure 3. It covers data pre-processing, further data augmentation, feature extraction with convolutional neural networks, training, data classification, evaluation metrics, and error analysis.

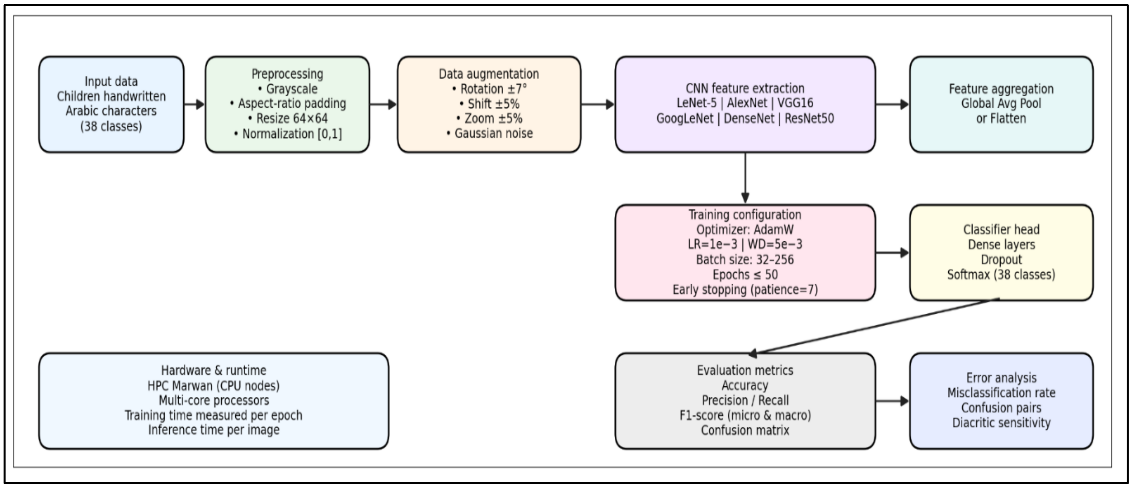
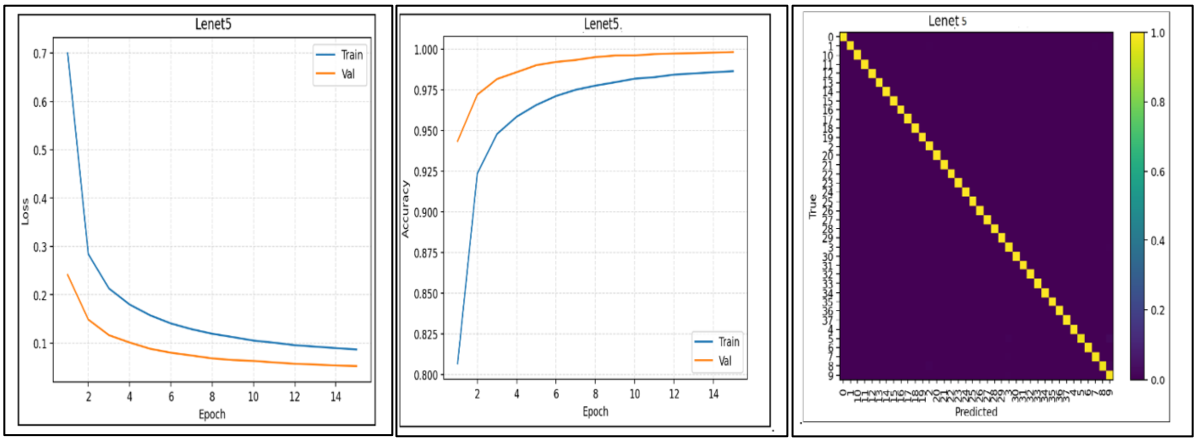


**Figure 3.** An Overview of Proposed CNN-Based Framework

## 4. Results and Discussion

The results in this section provide an accuracy and loss curve to evaluate the stability of learning, whereas confusion matrices highlight the ability of models to differentiate between visually similar classes. Computational metrics (GFLOPs, runtime per epoch) offer crucial feedback on the real-world effectiveness of the models. Together, they form a basis for understanding the value, advantages, and limitations of each architecture in practice. The following section discusses these results in terms of generalization, convergence rate, and computational accuracy.

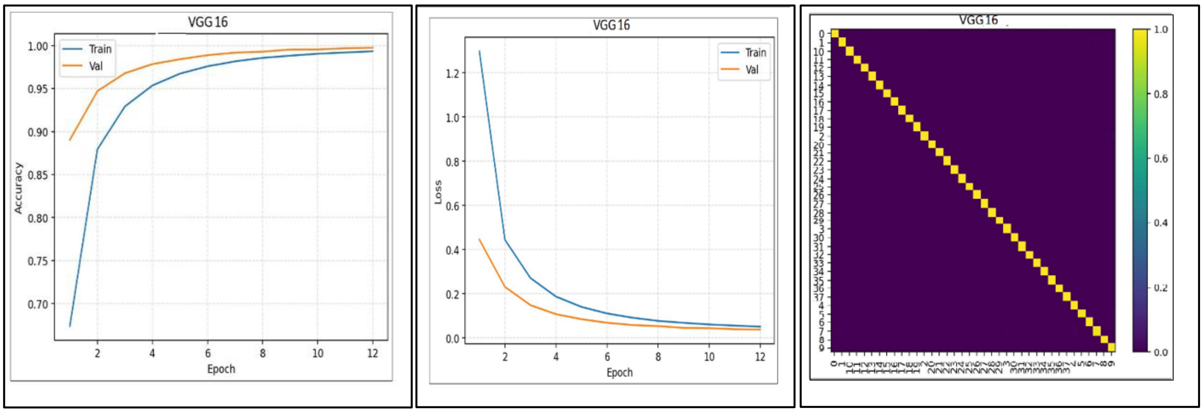
The LeNet-5 model offers stable learning, fast convergence, and excellent generalization. With a computational cost of around 0.0238 GFLOPs and a runtime of around 33 seconds per epoch, it represents a good compromise between performance and efficiency, notably for resource-limited environments. Subfigures 4(a), 4(b) and 4(c) below show the evolution of accuracy, the decrease in loss and the confusion matrix, which shows an almost perfect classification of handwritten classes, respectively.



(a). Accuracy Curve of LeNet5 (b). Loss Curve of LeNet5 (c). Confusion Matrix of LeNet5

**Figure 4.** Metrics of Accuracy, Loss and Confusion Matrix of LeNet5 Model

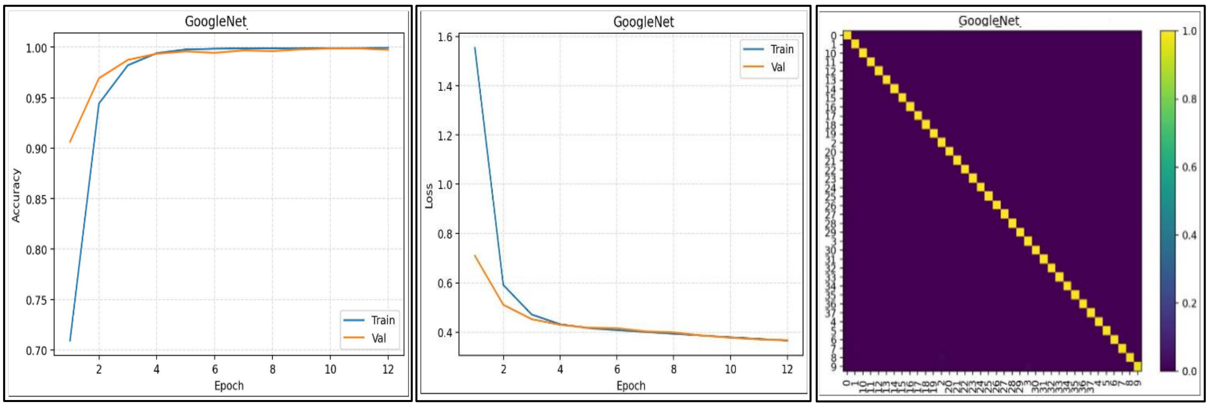
The VGG-16 model demonstrates stable learning, consistent convergence, and high generalization ability; however, it incurs high computational costs due to its depth. With a computational cost of approximately 1.05 GFLOPs and a runtime of approximately 42 seconds per epoch, it represents an attractive compromise between accuracy and computational performance. Subfigures 5(a), 5(b) and 5(c) provide a visual representation of the rapid progress in accuracy, the constant reduction in loss, and a perfectly organized confusion matrix.



(a). Accuracy Curve of VGG16 (b). Loss Curve of VGG16 (c). Confusion Matrix of VGG16

**Figure 5.** Metrics of Accuracy, Loss and Confusion Matrix VGG16 Model

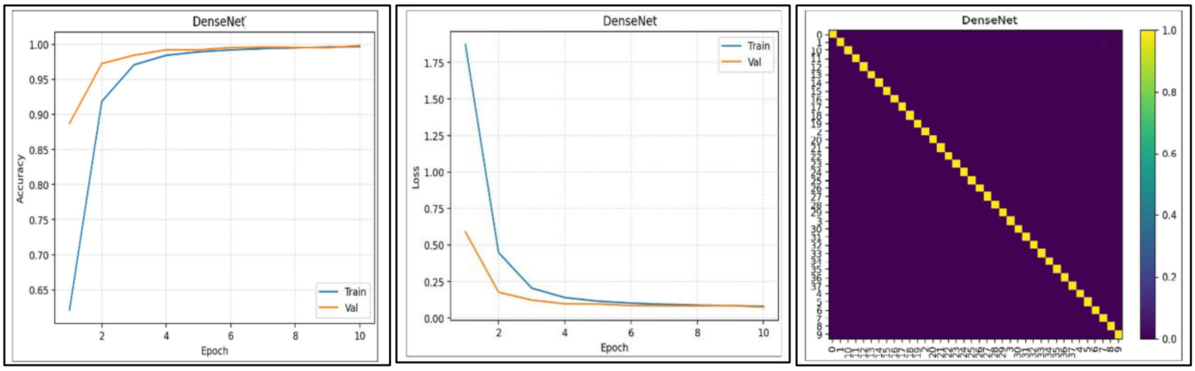
The GoogleNet model achieves rapid convergence and high stability to its flexible Inception architecture. It costs about 1.05 GFLOPs to run, and takes about 42 seconds per epoch to do so, which is a valid compromise between accuracy and speed. Subfigures 6(a), 6(b) and 6(c) illustrate near-linear accuracy progression, a constant decrease in loss, and a highly accurate confusion matrix. This suggests significant robustness to handwriting variations.



(a). Accuracy Curve of GoogLeNet (b). Loss Curve of GoogLeNet (c). Confusion Matrix of GoogLeNet

**Figure 6.** Metrics of Accuracy, Loss and Confusion Matrix GoogLeNet Model

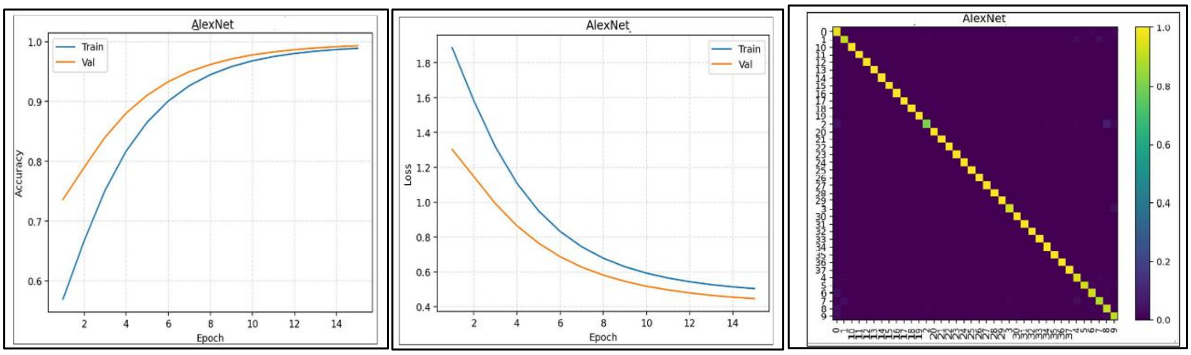
DenseNet offers stable learning through dense information flow between layers, which promotes rapid convergence and good generalization. With 1.28 GFLOPs and an execution time of approximately 44 seconds per epoch, it is one of the most powerful and robust models in the study. Subfigures 7(a), 7(b) and 7(c) below reveal its performance with smooth curves and a near-perfect confusion.



(a). Accuracy Curve of DenseNet (b). Loss Curve of DenseNet (c). Confusion Matrix of DenseNet

**Figure 7.** Metrics of Accuracy, Loss and Confusion Matrix of DenseNet Model

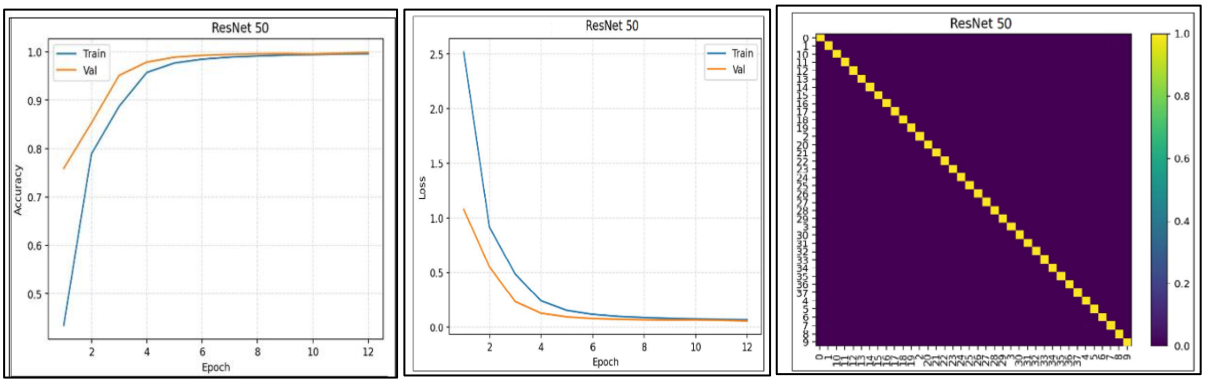
AlexNet has progressive and stable learning, but converges less quickly than more recent architectures. With a computational cost of 0.82 GFLOPs and an average execution time of 41 seconds per epoch, this model is a good compromise between efficiency and correctness, but suffers from a lack of generalization. Subfigures 8(a), 8(b) and 8(c) below highlight the improvement in accuracy, the decrease in loss and the confusion matrix with some persistent errors of this architecture.



(a). Accuracy Curve of AlexNet (b). Loss Curve of AlexNet (c). Confusion Matrix of AlexNet

**Figure 8.** Metrics of Accuracy, Loss and Confusion Matrix of AlexNet Model

ResNet50 exhibits learning stability, with faster convergence due to residual connections that improve gradient propagation. It has a computational load of 0.96 GFLOPs and an average execution time of 40 seconds per epoch, providing an ideal compromise between high performance, speed, and effectiveness. Figures 9(a), 9(b) and 9(c) below show almost perfect accuracy, very low loss and a perfect diagonal confusion matrix, confirming the superiority of this architecture.



(a). Accuracy Curve of ResNet50 (b). Loss Curve of ResNet50 (c). Confusion Matrix of ResNet50

**Figure 9.** Metrics of Accuracy, Loss and Confusion Matrix of ResNet50 Model

The observed training behavior does not indicate any bias in the model evaluation. Data augmentation and dropout regularization are applied only to the training set during training, which intentionally makes the optimization process more challenging. The validation set, on the other hand, is tested on clean augmented data without dropout, which naturally leads to a reduced validation loss.  To avoid biased assessment, stratified data splitting and fixed random seeds are used. Next, a set of tests is applied to verify the effectiveness of this method. These measures enhance the reliability and generalizability of the reported results.

A complete comparison of several convolutional neural network topologies (CNNs) shows that deeper models perform better than their predecessors. Table 4 and Figure 10 display the varied metrics and indicators of performance for all models. The ResNet-50 and DenseNet models are the best overall. GoogleNet's improved Inception architecture also makes it a nice balance between accuracy and speed. Although it is robust, VGG-16 suffers from parametric complexity and a higher GFLOPs computational load, limiting its performance compared to newer architectures.

AlexNet has good convergence but poor generalization, while LeNet-5 remains effective in low-power scenarios but is less accurate on more complex datasets.

In summary, empirical results indicate that residual and dense architectures are the most appropriate solutions for practical implementations that simultaneously demand high performance, speed and robustness.

The recognition error rate is an important measure of the reliability of a handwritten character recognition system. In this study, recognition error is clearly defined as the inverse of classification accuracy, meaning that recognition error = 1 − accuracy. The results in Table 4 show that the proposed framework has very low recognition error rates for models that perform well. For example, the ResNet50 model has an accuracy of 99.89%, which means that it makes a mistake only 0.11% of the time. On the other hand, DenseNet and GoogleNet have error rates of less than 0.15%. The low error rates demonstrate that the proposed architecture based on convolutional neural networks is robust. The majority of the remaining errors in recognition are caused by Arabic letters that seem identical, especially those that have distinct forms or strokes that are somewhat different. These are all factors that make it challenging to read what children write.

The experimental results in Table 4 indicate that the ResNet50 model outperforms all other CNN versions. It has an overall accuracy of 99.89% and an F1 score of 99.89%, and it consumes only 0.96 GFLOPs of computing power per image and takes approximately 40 seconds to run each epoch. The balancing act between accuracy and efficiency shows how effective residual learning is in avoiding degradation of gradient vectors and accelerating convergence of paths even without the use of the CPU. The LeNet5 model, on the other hand, achieved comparable accuracy with fewer parameters and a shorter runtime. In other words, it is a suitable choice for real-time or low-power embedded systems that prioritize computational efficiency over depth. AlexNet, on the other hand, did not have a normalization layer and had a fairly deep architecture, which slowed down convergence and affected its generalization ability. Overall, the results show that residual CNN architectures that are both lightweight and densely connected (ResNet50, DenseNet) offer the best compromise between accuracy, efficiency and complexity. This is a good starting point for further research on AI for the evaluation of bilingual education systems and for the recognition of children's Arabic handwriting.

**Table 4.** Comparison of the Proposed Models to Earlier Researches

| **Model** | **Parameters (M)** | **FLOPs (G)** | **Runtime Per epoch** | **Global Accuracy** | **Precision (Micro)** | **Recall (Micro)** | **F1(Micro)** | **Precision (Macro)** | **Recall (Macro)** | **F1(Macro)** |
|---|---|---|---|---|---|---|---|---|---|---|
| LeNet5 | 0,061 | 0,0238 | 33s | 99.85% | 99.85% | 99.85% | 99.85% | 99.86% | 99.86% | 99.86% |
| VGG16 | 1,2 | 1,22 | 37s | 99.75% | 99.75% | 99.75% | 99.75% | 99.76% | 99.76% | 99.76% |
| GoogleNet | 1,5 | 1,05 | 42s | 99.88% | 99.73% | 99.73% | 99.73% | 99.74% | 99.73% | 99.73% |
| AlexNet | 2,5 | 0,82 | 41s | 80.57% | 80.57% | 80.57% | 80.57% | 19.89% | 17.17% | 17.96 % |
| DenseNet | 0,81 | 1,28 | 44s | 99.86% | 99.86% | 99.86% | 99.86% | 99.86% | 99.86% | 99.86% |
| ResNet50 | 1,05 | 0,96 | 40s | 99.89% | 99.89% | 99.89% | 99.89% | 99.89% | 99.89% | 99.89% |

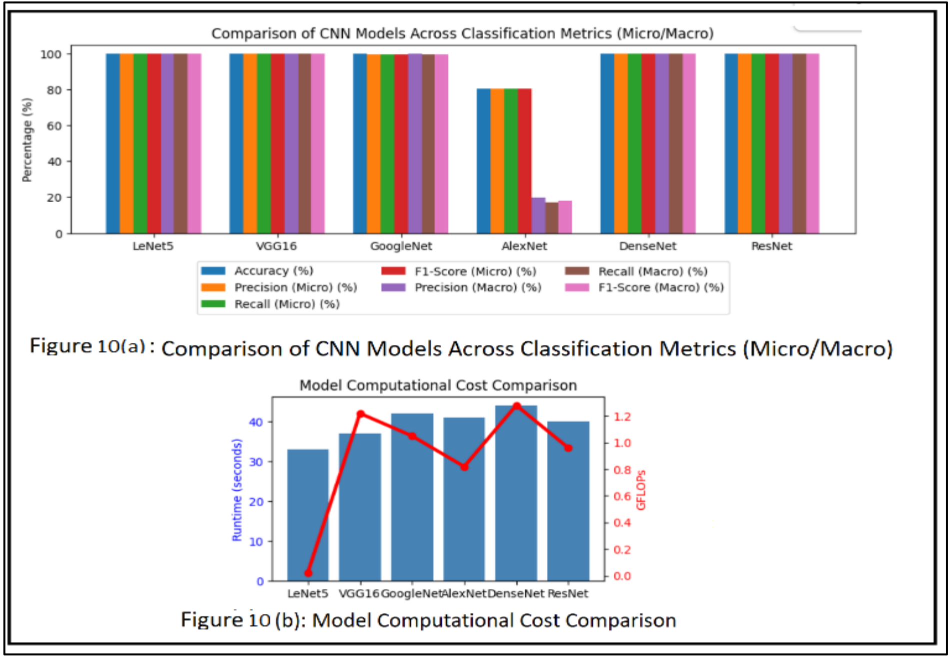


Figure 10(a): Comparison of CNN Models Across Classification Metrics (Micro/Macro)

Figure 10 (b): Model Computational Cost Comparison

**Figure 10.** Overall Comparison of Metrics and Performance Indicators Across All Model

All CNN architectures converged rapidly in less than 10 epochs. All models apart from AlexNet (81%) had validation accuracies above 99%; they learned even with only the CPU. Early stopping prevented the model from overfitting, and the training and validation losses came together perfectly. Computationally, validation losses were always lower than training losses for all models. This ensures that the optimization has been successful and that there has been no overfitting. Although AlexNet achieved a higher precise F1 score, its low overall F1 score is a result of inconsistent performance across classes and has nothing to do with the imbalance of the dataset. Although AlexNet excels on a limited set of classes, it is heavily penalized by macro-average metrics when it has to deal with visually complex characters.

The classification error rate is calculated by dividing the number of non-country entries in the confusion matrix by the total number of test samples. For the best-performing architectures, ResNet50, DenseNet, and GoogLeNet, the classification error rate remains below 0.2%. Consequently, misclassification cases remain rare. After closely examining the confusion matrices, it appears that most mistakes come from Arabic letters that distinguished, especially those that can only be told apart by slight changes in typefaces or diacritics. In this situation, the lines are not straight and are not in the correct location when trying to read children's handwriting. There is no common mistake among various sorts of letters, which leads to a mix-up of the correct representations of the qualities acquired with the visual aspects of the group.

The confusion matrix for ResNet50 was almost perfectly diagonal. That is, each class was well defined, and there was not much confusion between the 38 letters and Arabic numbers.

The new dataset used in this work (570,000 handwritten images of children) achieves much improved accuracy over previous work on Arabic OCR, which used either small corpora or only adult handwriting [9], [11], [15], [16]. Moreover, the combination of execution time and FLOPs reveals that, when architectural depth and normalizations are well adjusted, it is possible to achieve both high accuracy and computational efficiency. These results confirm the robustness and applicability of the proposed models for large-scale real-world applications.

The proposed CNN-based architecture works well overall; a closer examination of the evaluation criteria reveals some important aspects. The confusion matrices show that the most severe errors occur when Arabic letters that look identical. Especially when the only difference is in diacritical marks or small changes in strokes. In children's handwriting, difficulties such as instability of writing, incomplete strokes and incorrect positioning of diacritical marks are the main causes of this type of error. However, AlexNet performs much worse, especially when it comes to the macro-average F1 score. Even though the data in the balanced dataset are distributed uniformly across classes, AlexNet performs noticeably worse, particularly when it comes to the overall F1 score. This implies that AlexNet is not flexible enough to adjust to changing class characteristics and does not consistently discriminate between classes. The suggested method performs well overall, but there are a few drawbacks that need to be noted. The capture of extremely fine diacritical features is particularly challenging when 64×64 resolution images are used. The second argument is that the models were trained and tested on just one data set. This might create a bias that only affects this set of data, making it hard to utilize the results for individuals of different ages or writing styles. Finally, samples with very poor or hard-to-read handwriting are still difficult to work with combining contextual or sequential models would further enhance the overall performance of the system.

## 5. Conclusion

The study indicates that when Arabic handwritten data is modified and then identified in a CPU-intensive process, constructs such as simplified CNNs can offer remarkable recognition and evaluation at exceptionally high standards. The accuracy complex with the best object computation is the ResNet50 structure, based on the results of the experiment. The process is considered effective for the majority of applications in handwriting recognition that deal with large quantities of data. However, in the case of simple models based on LeNet-5, faster inferencing results suggest that they would be appropriate for real-time applications that require speed. The higher-quality projects are more efficient, despite the higher running costs, according to comparative calculations. On the other hand, lightweight CNNs have deployment ease and low training and interpretive costs. This benchmark examines and advises on the best architecture for educational activities, compact gadgets, or any other case that only requires central controllers. While the overall performance is excellent, there are still small errors, mainly on letters with few or no graphic characteristics. Consequently, this result indicates that the handwriting recognition engine development workflow should include more analysis of the errors and field considerations.

Further work focusing on the development of the given approach in different directions is needed. To begin with, it might be possible to combine feature extractions based on CNNs with a transformer-based architecture or an attention-driven architecture to make them less sensitive to long-range dependency and varying handwriting styles. Second, it is important to explore ways of making explainable AI technologies more digestible and allowing individuals to make informed choices in relation to education. Additionally, the analysis of the system using writer-independent and age-independent data, along with the investigation of inputs with high precision and location for device usage, can be considered exciting options for future study.

## References

[1] Chakrani, Brahim, Adam Ziad, and Abdenbi Lachkar. "Paradoxes of Language Policy in Morocco: Deconstructing the Ideology of Language Alternation and the Resurgence of French in STEM Instruction." Languages 10, no. 6 (2025): 135.

[2] Alenezi, Mamdouh. "Digital Learning and Digital Institution in Higher Education." Education Sciences 13, no. 1 (2023): 88.

[3] Altwaijry, Najwa, and Isra Al-Turaiki. "Arabic Handwriting Recognition System Using Convolutional Neural Network." Neural Computing and Applications 33, no. 7 (2021): 2249-2261.

[4] AlKendi, Wissam, Franck Gechter, Laurent Heyberger, and Christophe Guyeux. "Advancements and Challenges in Handwritten Text Recognition: A Comprehensive Survey." Journal of Imaging 10, no. 1 (2024): 18.

[5] Alrasheed, Ghady, and Suliman A. Alsuhibany. "Enhancing Security of Interfaces: Adversarial Handwritten Arabic CAPTCHA Generation." Applied Sciences 15, no. 6 (2025): 2972.

[6] El Khayati, Mohsine, Ismail Kich, and Youssef Taouil. "CNN-Based Methods for Offline Arabic Handwriting Recognition: A Review." Neural Processing Letters 56, no. 2 (2024): 115.

[7] KO, Mohammed Aarif, and Sivakumar Poruran. "OCR-Nets: Variants of Pre-Trained CNN for Urdu Handwritten Character Recognition Via Transfer Learning." Procedia computer science 171 (2020): 2294-2301.

[8] Gupta, Deepika, and Soumen Bag. "CNN-based Multilingual Handwritten Numeral Recognition: A Fusion-Free Approach." Expert Systems with Applications 165 (2021): 113784.

[9] Alwagdani, Maram Saleh, and Emad Sami Jaha. "Deep Learning-Based Child Handwritten Arabic Character Recognition and Handwriting Discrimination." Sensors 23, no. 15 (2023): 6774.

[10] Almuhaideb, Sarab, Najwa Altwaijry, Ahad D. Alghamdy, Daad Alkhulaiwi, Raghad Alhassan, Haya Alomran, and Aliyah M. Alsalem. "Dhad—A Children's Handwritten Arabic Characters Dataset for Automated Recognition." Applied Sciences 14, no. 6 (2024): 2332.

[11] Michalski, Bartosz & Plechawska-Wójcik, Małgorzata. (2022). Comparison of LeNet-5, AlexNet and GoogLeNet Models in Handwriting Recognition. Journal of Computer Sciences Institute. 23. 145-151. 10.35784/jcsi.2919.

[12] Niharmine, Lahcen, Benaceur Outtaj, and Ahmed Azouaoui. "Tifinagh Handwritten Character Recognition Using Optimized Convolutional Neural Network." International Journal of Electrical & Computer Engineering (2088-8708) 12, no. 4 (2022).

[13] Ladrham, Khalil, and Hicham Gueddah. "Advanced OCR for Digits Exploring CNN for Optimal Performance." Procedia Computer Science 251 (2024): 734-739.

[14] Alsayed, Alhag, Chunlin Li, Ahmed Fat'hAlalim, Mohammed Hafiz, Jihad Mohamed, Zainab Obied, and Mohammed Abdalsalam. "The Impact of Various Factors on the Convolutional Neural Networks Model on Arabic Handwritten Character Recognition." International Journal of Advanced Computer Science & Applications 15, no. 5 (2024).

[15] Ali, Sardar Hasen, and Maiwan Bahjat Abdulrazzaq. "KurdSet Handwritten Digits Recognition Based on Different Convolutional Neural Networks Models." TEM Journal 13, no. 1 (2024).

[16] Finjan, Rasool Hasan, Ali Salim Rasheed, Ahmed Abdulsahib Hashim, and Mustafa Murtdha. "Arabic Handwritten Digits Recognition Based on Convolutional Neural Networks with Resnet-34 Model." Indonesian Journal of Electrical Engineering and Computer Science 21, no. 1 (2021): 174-178.

[17] Al-Maamari, Mohammed R., Rakesh Ramteke, Aymen M. Al-Hejri, and Sultan S. Alshamrani. "Integrating CNN and Transformer Architectures for Superior Arabic Printed and Handwriting Characters Classification." Scientific reports 15, no. 1 (2025): 29936.

[18] Hussain, Nadir, Mushtaq Ali, Sidra Abid Syed, Rania M. Ghoniem, Nazia Ejaz, Omar Imhemed Alramli, Mohammed Alaa Ala'anzy, and Zulfiqar Ahmad. "Design and Evaluation of Arabic Handwritten Digit Recognition System Using Biologically Plausible Methods." Arabian Journal for Science and Engineering 49, no. 9 (2024): 12509-12523.

[19] Alghyaline, Salah. "Optimised CNN Architectures for Handwritten Arabic Character Recognition." Computers, Materials & Continua 79, no. 3 (2024).

[20] Krichen, M. (2023). Convolutional Neural Networks: A Survey. Computers, 12(8), 151. https://doi.org/10.3390/computers12080151

[21] Nugraha, Gibran Satya, Muhammad Ilham Darmawan, and Ramaditia Dwiyansaputra. "Comparison of CNN’s Architecture GoogleNet, AlexNet, VGG-16, Lenet-5, Resnet-50 in Arabic Handwriting Pattern Recognition." Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control (2023).

[22] Mutawa, A. M., Mohammad Y. Allaho, and Monirah Al-Hajeri. "Machine Learning Approach for Arabic Handwritten Recognition." Applied Sciences 14, no. 19 (2024): 9020.