applied sciences

MDPI

Article

Dhad—A Children's Handwritten Arabic Characters Dataset for Automated Recognition

Sarab AlMuhaideb *, Najwa Altwaijry, Ahad D. AlGhamdy, Daad AlKhulaiwi, Raghad AlHassan, Haya AlOmran and Aliyah M. AlSalem

Computer Science Department, College of Computer and Information Sciences, King Saud University, Riyadh 11362, Saudi Arabia; ntwaijry@ksu.edu.sa (N.A.); 441201150@student.ksu.edu.sa (A.D.A.); 441201060@student.ksu.edu.sa (D.A.); 441201024@student.ksu.edu.sa (R.A.); 441200957@student.ksu.edu.sa (H.A.); 441203529@student.ksu.edu.sa (A.M.A.)
* Correspondence: salmuhaideb@ksu.edu.sa

Abstract: This study delves into the intricate realm of recognizing handwritten Arabic characters, specifically targeting children's script. Given the inherent complexities of the Arabic script, encompassing semi-cursive styles, distinct character forms based on position, and the inclusion of diacritical marks, the domain demands specialized attention. While prior research has largely concentrated on adult handwriting, the spotlight here is on children's handwritten Arabic characters, an area marked by its distinct challenges, such as variations in writing quality and increased distortions. To this end, we introduce a novel dataset, "Dhad", refined for enhanced quality and quantity. Our investigation employs a tri-fold experimental approach, encompassing the exploration of pre-trained deep learning models (i.e., MobileNet, ResNet50, and DenseNet121), custom-designed Convolutional Neural Network (CNN) architecture, and traditional classifiers (i.e., Support Vector Machine (SVM), Random Forest (RF), and Multilayer Perceptron (MLP)), leveraging deep visual features. The results illuminate the efficacy of fine-tuned pre-existing models, the potential of custom CNN designs, and the intricacies associated with disjointed classification paradigms. The pre-trained model MobileNet achieved the best test accuracy of 93.59% on the Dhad dataset. Additionally, as a conceptual proposal, we introduce the idea of a computer application designed specifically for children aged 7–12, aimed at improving Arabic handwriting skills. Our concluding reflections emphasize the need for nuanced dataset curation, advanced model architectures, and cohesive training strategies to navigate the multifaceted challenges of Arabic character recognition.

Keywords: deep learning; pre-trained models; child handwriting recognition; Dhad; Hijja

Citation: AlMuhaideb, S.; Altwaijry, N.; AlGhamdy, A.D.; AlKhulaiwi, D.; AlHassan, R.; AlOmran, H.; AlSalem, A.M. Dhad—A Children's Handwritten Arabic Characters Dataset for Automated Recognition. *Appl. Sci.* **2024**, *14*, 2332. https://doi.org/10.3390/app14062332

Academic Editors: Zahid Mehmood Jehangiri, Mohsin Shahzad and Uzair Khan

Received: 3 January 2024
Revised: 3 March 2024
Accepted: 8 March 2024
Published: 10 March 2024

1. Introduction

Arabic is a widely spoken language, with over 360 million people using it as their primary language [1]. In the domain of language processing and technological applications, the recognition of handwritten Arabic characters, especially in the realm of children's script, poses unique challenges [2,3]. Arabic, being a Semitic language, presents inherent complexities in its script, demanding advanced algorithms for precise recognition. These challenges emanate from factors such as the semi-cursive nature of Arabic writing, distinct character shapes based on their position in a word, and the presence of diacritical marks representing short vowels and other phonetic features. The significance of addressing the issue of Arabic handwritten recognition, particularly concerning children, arises from the increasing integration of technology in their educational and recreational activities. The ubiquitous use of smartphones and tablet devices by children, employing touchscreens and styluses for various purposes, including handwriting, underscores the need for automated recognition techniques tailored to the unique characteristics of children's handwriting [4–6].

While the existing literature has made notable strides in Arabic handwritten recognition, the focus has predominantly been on adult handwriting. Researchers have explored

diverse datasets such as the Arabic Handwritten Characters Dataset (AHCD) [7] and the Database of Arabic Handwritten Characters and Ligature (DBAHCL) [8], achieving commendable accuracy rates using both conventional methods (e.g., Support Vector Machine (SVM), K-Nearest Neighbour (KNN)) and advanced techniques (e.g., Convolutional Neural Network (CNN) and Artificial Neural Network (ANN)). Notably, CNNs [9,10] have emerged as powerful tools for feature extraction, demonstrating superior performance compared to traditional machine learning approaches. In the context of children's handwriting recognition, the existing Hijja dataset [4,11] is the only resource facilitating the training of deep learning classification models. However, children's handwriting introduces additional complexities, including variations in writing quality, increased variances, and more substantial distortions. Recognizing these distinctions is imperative for developing effective applications in education, interactive learning, and other practical domains tailored to children.

Limited research has been conducted in recognizing children's written Arabic characters using the Hijja dataset. The existing literature uses conventional and deep learning approaches towards the classification of children written Arabic characters. Altwaijry and Al-Turaiki [4] introduced the unique Hijja dataset, training a CNN model with 88% accuracy, but lacked detailed investigations into existing powerful models. Alkhateeb et al. [12] implemented a custom CNN model, achieving 92.5% accuracy on the Hijja dataset. Nayef et al. [13] introduced the Optimized Leaky Rectified Linear Unit (OLReLU)-CNN model, attaining 90% accuracy on Hijja. Alwagdani and Jaha [5] explored custom CNN models, emphasizing the impact of diverse training datasets and achieving an impressive average accuracy of 92.78% on recognizing children's handwritten characters, while also proposing supplementary features for enhanced discrimination. Alheraki et al. [14] tailored a custom CNN for achieving 91% accuracy on Hijja. Recently, Bin Durayhim et al. [15] implemented a custom CNN and pre-trained VGG16 models, reporting a remarkable 99% accuracy on the Hijja dataset and introducing the Mutqin application for children's practice. These studies collectively highlight the evolving landscape of deep learning applications in recognizing children's Arabic handwriting. However, concerns about model generalization and sensitivity persist, and further exploration is warranted in this context.

This paper introduces a new dataset, "Dhad", following procedures similar to those for Hijja to ensure consistency. The Dhad dataset features improved sample quality, enhanced preprocessing to remove noisy elements, and a greater number of samples. This manuscript systematically addresses the problem by investigating the potential of existing pre-trained powerful CNN models using the transfer learning technique. Furthermore, it explores the performance of simpler CNN models trained from scratch and classification on deep visual features. In summary, the anticipated contributions of this manuscript include the following:

1. Introduction of the new "Dhad" dataset to facilitate the training of deep learning models for children's handwritten Arabic characters.
2. Investigation of the potential of pre-trained powerful CNN models for children's handwritten Arabic character classification.
3. Examination of the performance of a simple CNN model trained from scratch for children's handwritten Arabic character classification.
4. Exploration of the classification performance on CNN-extracted features using conventional machine learning models including SVM and Random Forest (RF).
5. Discussion of the practical use-case of the trained classification model, emphasizing the potential utility of children's handwritten Arabic characters recognition.

2. Background to Deep Learning Models

2.1. ResNet50

He et al. [16] introduced an innovative approach to training highly deep neural networks, proposing a novel framework based on residual learning. Instead of training networks to learn unreferenced functions, the authors suggested a reinterpretation of layers

as residual learning functions by referencing the input of the layer. This concept of residual learning played a crucial role in the optimization of deep networks, enabling the attainment of enhanced accuracy with deep models. To express this mathematically, if we denote the desired mapping function as $H(x)$, in the context of residual learning, stacked non-linear layers are designed to fit another mapping function $F(x) := H(x) - x$, where x represents the input to the layer. This approach significantly contributed to the effectiveness of training deep networks by explicitly capturing the residual information between the desired and actual mappings.

2.2. MobileNet

Howard et al. [17] introduced a new type of CNN called MobileNets, tailored for high-performance applications on advanced hardware. The key innovation involves using depth-wise separable convolutions to efficiently build deep networks. This method introduces two global hyperparameters, allowing customization for specific problems while balancing accuracy and latency. Depth-wise separable convolution, a specialized type of convolution, breaks down the standard convolution process into two steps. First, a depth-wise convolution is applied, and then, a 1×1 point-wise convolution combines the results from the previous layer. Importantly, each layer in the network is followed by BatchNormalization and Rectified Linear Unit (ReLu) non-linearity. This separation into depth-wise and point-wise convolutions helps improve computational efficiency while maintaining the network's performance.

2.3. DenseNet121

Huang et al. [18] introduced DenseNet, a novel class of densely connected convolutional networks that builds upon the idea of residual connections present in traditional networks. The key innovation involves establishing connections from each layer to every other layer in the feedforward direction. This architectural choice means that each layer receives the feature maps of all preceding layers as input, resulting in a network with $L(L+1)/2$ connections, in contrast to the L connections in traditional networks with L layers. The advantages of densely connected networks include improved feature propagation, efficient feature reuse, a substantial reduction in the number of network parameters, and mitigation of the vanishing-gradient problem. Unlike the approach in residual networks, where feature maps are added before feeding into the next layer, DenseNet combines feature maps through concatenation. Mathematically, if a network comprises L layers, each with a non-linear transformation represented by a composite function F_l, the output x_l for the densely connected layer can be understood as the concatenation of feature maps from the previous layers. In practical terms, this means that each layer's output includes information from all preceding layers, promoting rich information flow and enhancing the network's ability to learn complex representations.

2.4. Custom CNN

A custom CNN model was developed, motivated from the literature [4,5,13–15] to investigate the performance of a simpler network trained from scratch for the children's handwritten character classification. The model is particularly tailored for grayscale images with a size of 32×32 pixels (Figure 1). The architecture is constructed as a sequential stack of layers using TensorFlow and Keras packages. The initial layer applies 64 convolutional filters of the size 3×3 with ReLU activation and the "same" padding, preserving spatial dimensions. Subsequent max pooling layers with 2×2 pool sizes and strides of two reduce the spatial dimensions. This pattern repeats with increased filter counts in deeper convolutional layers (128 and 256 filters). Dropout layers with a rate of 0.3 are strategically inserted to mitigate overfitting. Following flattening, two densely connected layers with 512 and 1024 units, respectively, deploy ReLU activation. The output layer, activated by softmax, consists of 29 units, aligning with the classification task's classes. The model is

compiled using the Adam optimizer, categorical cross-entropy as the loss function, and accuracy as the performance metric.

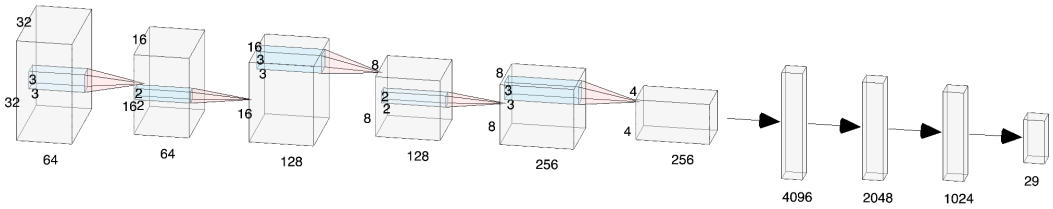


Figure 1. The architecture of our custom CNN model.

3. Related Work

This section presents a summary of the literature in the context of children's handwriting classification. First, a brief overview of how technology evolved for the children's handwriting classification is presented. In the second section, a more targeted review of more recent deep learning-based research in the context of children's written Arabic character classification is provided to demonstrate the state of the art in this domain.

3.1. Children's Handwriting Classification

The classification of children's handwriting has been an active area of research for decades, driven by the need for objective and automated assessment tools. Unlike adult handwriting, which tends towards standardization, children's writing exhibits a wide range of variations due to age, developmental stage, and individual learning styles. Early attempts at automated character recognition (OCR) for children's handwriting often relied on template matching techniques. Pioneering works employed pre-defined templates representing ideal character shapes. An input character would be compared to these templates, with the closest match assigned as the recognized character. However, this approach proved ineffective for children's handwriting due to its inherent lack of conformity [19,20]. However, there are a number of limitations identified in the template matching approaches, particularly for characters with significant variations in form.

Researchers recognized the limitations of template matching and explored alternative approaches. Utilizing statistical and structural features for character classification has been explored by researchers [21–23]. This involved analysing features like line endings, crossings, and loops within handwritten characters. While offering more flexibility than rigid templates, these methods still faced limitations. The emergence of deep learning techniques, particularly CNNs, has revolutionized the field. Unlike previous methods, CNNs excel at extracting intricate features from the data. This allows them to effectively capture the natural variations in children's handwriting, leading to more robust and accurate character-level classification. Further advancements in deep learning architectures, such as recurrent neural networks (RNNs), have shown promise in handling the sequential nature of handwriting data.

3.2. Deep Learning-Based Classification of Children's Arabic Handwriting

A summary of the latest research related to children's handwritten Arabic character classification is presented in this section to demonstrate the state of the art. This section is organized in chronological order to better understand the developments over the years.

Alkhateeb et al. [12] in 2020 implemented a custom CNN model for the classification of Arabic characters using the AHCR, AHCD, and Hijja datasets. The authors reported an accuracy of 92.5% for the Hijja dataset. Altwaijry and Al-Turaiki [4] in 2021 introduced the Hijja dataset, which stands out as a unique collection focusing exclusively on letters written by children. This dataset, comprising 47,434 characters from 591 participants aged 7–12, filled a notable gap in existing resources, particularly for understanding the nuances of children's handwriting. A CNN model was developed and trained on the Hijja dataset, which achieved a test accuracy of 88%. However, the work lacked a detailed investigation on existing powerful models and ignored the practical implications of the research.

Nayef et al. [13] in 2022 introduced an OLReLU combined with a CNN architecture and a batch normalization layer to enhance performance in scenarios with imbalanced positive and negative vectors. Four datasets, including the AHCD, self-collected data, Modified National Institute of Standards and Technology (MNIST), and AlexU Isolated Alphabet (AIA9K), were used. The proposed model was able to achieve 90% accuracy for the Hijja dataset. Alwagdani and Jaha [5] recently in 2023 investigated the problem in more detail using custom developed CNN models and hybrid approaches. The authors made use of datasets from both adults (i.e., AHCD) and children (i.e., Hijja) to explore the performance, with a particular emphasis on the impact of different training datasets. The authors further investigated the problem of classifying by deploying a conventional machine learning pipeline of extracting visual features and classifying using classical machine learning models (e.g., SVM, KNN, and RF). The findings reveal that training the model on a combination of children's and adult datasets yields the best performance, achieving an impressive average accuracy of 92.78% in recognizing children's handwritten characters. Moreover, authors extended their investigation to the classification of writers into two groups (i.e., children and adults) using the proposed CNN model. The initial results showed an average accuracy of 89.28%, indicating the presence of confusable similarities in writing styles between adults and children. To enhance discrimination performance, the study suggested supplementary features based on Histogram of Oriented Gradients (HOGs) and statistical measures, which, when combined with CNN features, result in a significantly improved accuracy of 92.29%.

Alheraki et al. [14] in 2023 implemented a custom CNN architecture tailored for recognizing children's Arabic handwritten characters. The authors made use of the AHCD and Hijja datasets to train the model and achieved accuracies of 97% and 91% for AHCD and Hijja, respectively. Additionally, the authors introduced an innovative approach using character strokes as a filter to further enhance recognition accuracy. This method, combined with CNNs, demonstrated effectiveness in improving performance. The research compared the proposed model with the pre-trained EfficientNetV0 and reported better performance for the custom model. Moreover, a multi-model approach, integrating information about the number of strokes in a character, achieved an average prediction accuracy of 96% when Hijja was merged with AHCD. Bin Durayhim et al. [15] in 2023 implemented two deep learning-based models (i.e., custom CNN and pre-trained VGG16) for children's handwriting character recognition using Hijja and AHCD. The custom CNN model was reported to outperform the VGG16 model and other models from the literature, achieving 99% accuracy on the Hijja dataset. Additionally, the paper introduced Mutqin, a prototype tablet application designed for children to practice Arabic handwriting and spelling, incorporating the best-performing CNN model. The application was evaluated through user acceptance testing, considering effectiveness, efficiency, and satisfaction, with positive results indicating good performance.

A notable progress has been made in the literature in regard to addressing the unique challenges posed by children's handwriting. Several studies have implemented various deep learning models to recognize isolated Arabic characters, particularly targeting children's writing styles. The introduced models include custom CNNs, OLReLU with CNN architectures, pre-trained models (i.e., VGG16 and EfficeintNetv0) and hybrid approaches combining CNNs with classical machine learning models (e.g., SVM, KNN, and RF). These studies have utilized datasets such as the AHCD and the specifically designed Hijja dataset, which exclusively feature letters written by children. The accuracy reported in these works range from 88% to 99%, showcasing the effectiveness of these models in handling the intricacies of children's handwriting.

However, certain limitations and gaps persist in the current research landscape. Notably, there is a lack of consistent exploration and discussion of transfer learning models, specifically ResNet50 and MobileNet, which have demonstrated success in other domains. Additionally, studies report inconsistent performance for almost similar CNN structures, indicating a potential non-reliability of simple CNN models for this specific problem. There

is no justification in the literature with regard to significantly varied performances for almost same CNN architectures with minor hyperparamter variations. Furthermore, the practical applications of the proposed models are not extensively discussed across the studies, with only one work introducing a prototype tablet application named Mutqin for children's practice.

4. Dhad Dataset Formation

In this section, we describe our collected dataset, Dhad (the dataset is available at https://github.com/daadturki1/Dhad/ (accessed on 1 October 2023)), and the steps taken to preprocess and prepare the dataset for the training of deep learning models. The Dahd dataset collection process was based on the procedures described by Altwaijry and Al-Turaiki [4]. The dataset was collected from Arabic-speaking school children between 7 and 12 years old within the Riyadh region in 2019. In total, 55,587 samples for all 29 letter classes in all different forms were collected. The count of each collected letter in all its forms after discarding noisy input is reported in Table 1. In our image processing workflow, the handwritten letters without dots were identified and cropped using the `findContours()` method available in the OpenCV library [24]. Specifically, this method was utilized to locate the outer contour of each object present in the image, after which we proceeded to crop the image around the largest identified contour. Conversely, for handwritten letters containing dots, the `findNonZero()` method was employed to identify all black pixels in the image. Subsequently, the smallest possible rectangle that encompassed all black pixels was cropped to isolate the desired letter. After that, the images were resized to 32×32 pixels. To eliminate the noise within the scanned images, we used a Gaussian filter with a kernel of size 5×5 to blur specific portions of the image. Then, a high-pass filter was used to sharpen the edges. Lastly, the binarization technique was used to convert the RGB image to a binary level with a global thresholding algorithm. Figure 2 depicts sample images from the classes "mīm" and "nūn", showing the different preprocessing stages. Four data augmentation techniques were used with a range of 0.2: height and width shift, shear range, zoom range, and rescale. The dataset was then normalized, shuffled, and split into 60%, 20%, and 20% for training, validation, and testing, respectively. Overall, 30,922, 10,300, and 10,333 samples were used for the training, validation, and testing sets, respectively. Figure 3 presents the dataset collection workflow.

Table 1. The different letter forms and the total number of images for each class after data cleansing in the collected dataset.

No.	Class	Form	Count	No.	Class	Form	Count
1	ʾalif	ا، أ، إ، ـأ، ـإ	2869	16	ṭāʾ	ط، طـ ، ـطـ ، ـط	1925
2	bāʾ	ب، بـ ، ـبـ ، ـب	1899	17	ẓāʾ	ظ، ظـ ، ـظـ ، ـظ	1886
3	tāʾ	ت، تـ ، ـتـ ، ـت	1920	18	ʿayn	ع، عـ ، ـعـ ، ـع	1906
4	ṯāʾ	ث، ثـ ، ـثـ ، ـث	1734	19	ġayn	غ، غـ ، ـغـ ، ـغ	2012
5	ǧīm	ج، جـ ، ـجـ ، ـج	1891	20	fāʾ	ف، فـ ، ـفـ ، ـف	2024
6	ḥāʾ	ح، حـ ، ـحـ ، ـح	1921	21	qāf	ق، قـ ، ـقـ ، ـق	2030
7	khāʾ	خ، خـ ، ـخـ ، ـخ	1869	22	kāf	ك، كـ ، ـكـ ، ـك	2019
8	dāl	د، ـد	931	23	lām	ل، لـ ، ـلـ ، ـل	2011
9	ḏāl	ذ، ـذ	915	24	mīm	م، مـ ، ـمـ ، ـم	1955
10	rāʾ	ر، ـر	898	25	nūn	ن، نـ ، ـنـ ، ـن	1913
11	zāy	ز، ـز	944	26	hāʾ	ه، هـ ، ـهـ ، ـه	2180
12	sīn	س، سـ ، ـسـ ، ـس	1845	27	wāw	و، ـو	872
13	šīn	ش، شـ ، ـشـ ، ـش	1876	28	yāʾ	ي، يـ ، ـيـ ، ـي	1903
14	ṣād	ص، صـ ، ـصـ ، ـص	1709	29	hamzah	ء، ؤ، ئ، ـئ	1846
15	ḍād	ض، ضـ ، ـضـ ، ـض	1852				
						Total	51,555

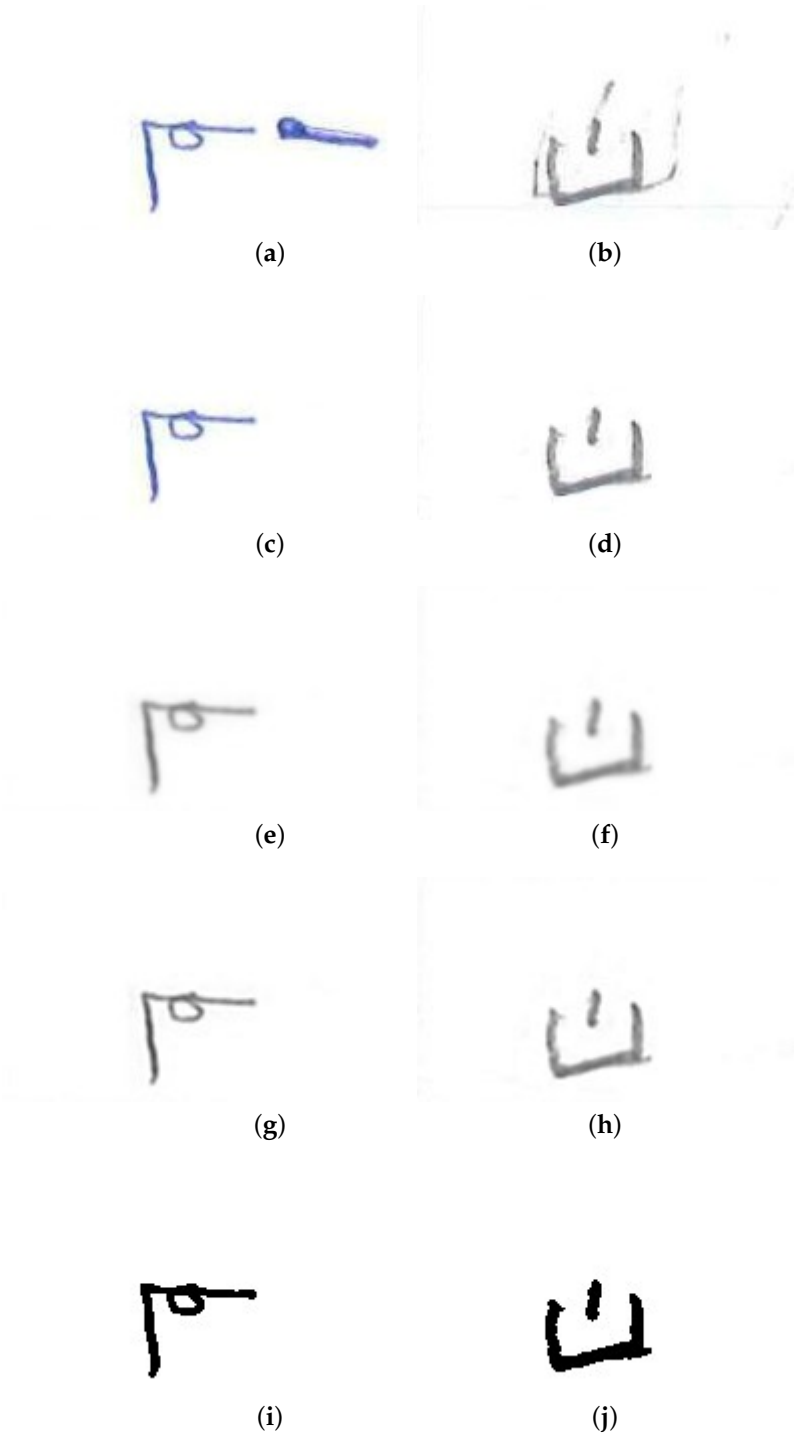


Figure 2. Sample images from the letters mīm and nūn showing the consecutive data cleansing and preprocessing steps. (**a**) Letter mīm after scanning and cropping. (**b**) Letter nūn after scanning and cropping. (**c**) Letter mīm after cleansing. (**d**) Letter nūn after cleansing. (**e**) Letter mīm after applying a Gaussian filter. (**f**) Letter nūn after applying the Gaussian filter. (**g**) Letter mīm after applying a high-pass filter. (**h**) Letter nūn after applying a high-pass filter. (**i**) Letter mīm after applying binarization. (**j**) Letter nūn after binarization.

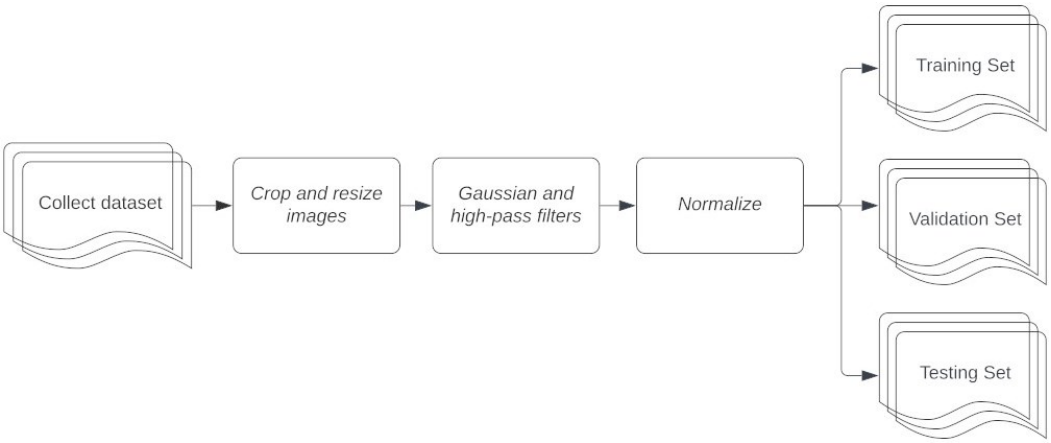


Figure 3. Dhad dataset collection and preparation workflow.

5. Experimental Design

To investigate the problem in details, in total, three experiments were performed:

- Experiment One—Pre-Trained Models: The first experiment was designed to explore the potential of existing pre-trained powerful CNN models in classifying children's written Arabic characters. The literature suggests that well-established CNN models pre-trained with large image datasets like that of ImageNet perform superior in comparison to training from scratch. In this context, the ResNet50, MobileNet, and DenseNet121 models are implemented for both the Hijja and Dhad datasets.
- Experiment Two—Custom CNN Model: The second experiment was designed to investigate the performance of a simpler custom CNN model for this problem. The development of custom CNN models has already been reported in the literature; however, varied performances are reported each time. In this experiment, we developed a simple CNN model inspired from the literature and implemented it for both the Hijja and Dhad datasets.
- Experiment Three—Classification on Deep Visual Features: The third experiment was designed to study the performance of classical classification models including SVM, RF, and MLP trained on deep visual features extracted by a deep learning CNN model (i.e., MobileNet trained over ImageNet and MobileNet trained in Experiment One).

6. Experimental Protocols and Evaluation Measures

We employed the OpenCV–4.9.0 library and the Python Imaging Library (Pillow 10.2.0) in our study, as they provide the necessary functions for data preprocessing and data augmentation and used the TensorFlow 2 [25] and Keras 3 [26] Python libraries to implement the architectures. We also used Google Colaboratory Pro [27] to speed up the training by granting access to K80, P100, T4 GPU, and 32 GB RAM. Gridsearch [28] was used for hyperparameter tuning. A dataset split of 60:20:20 was used for training, validation, and testing purposes using the conventional hold-out approach. All the pre-trained models were fine-tuned for 30 epochs, while the custom CNN model was trained from scratch for 100 epochs to ensure convergence. For all the models, the Adam optimizer was used with categorical cross-entropy loss, and a training batch size of 4 was used.

The performance of the trained models was evaluated for both the training and validation phases using standard evaluation measures. The training performance was assessed based on the training loss curves, validation loss curves, training accuracy curves, and validation accuracy curves. Further, for the best epoch model based on the validation loss, the results for all four measures were also reported in tabular format. The testing performance was assessed using test accuracy, test loss, F1 score, precision score, recall score, and J-index. In addition to these quantitative measures, confusion matrices, and Area Under Curve (AUC) curves were used to analyse the class-wise performances of

trained models. For the pre-trained models, to better understand the performance, layer visualizations were also plotted.

Let TP and TN denote the numbers of true positives and true negatives, respectively, and FP and FN denote the numbers of false positives and false negatives, respectively. Accuracy (Acc) is defined as the proportion of correctly predicted examples (1). The loss ($Loss$) quantifies the degree of misclassification by determining the proportion of incorrect predictions relative to the total predictions made by the model (2). Precision (P) is the fraction of correctly classified positive examples among all positively classified examples (3). Meanwhile, recall or sensitivity measures the ratio of correctly classified positive examples to the true positive examples (4). The $F1$ score is calculated as the harmonic mean of the precision and recall; thus, it combines both precision and recall in a single value (5). The Jaccard Similarity Index (often referred to simply as $J\text{-}Index$) is a measure of how close the predicted labels are to the actual labels (6).

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Loss = \frac{FP + FN}{TP + TN + FP + FN} \tag{2}$$

$$P = \frac{TP}{TP + FP} \tag{3}$$

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

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{5}$$

$$J\text{-}Index = \frac{TP}{TP + FP + FN} \tag{6}$$

7. Results

In addressing the challenges of handwritten Arabic characters among children, a series of experiments were conducted and their outcomes are presented in this section. The results encompass both numerical evaluations and graphical representations. To offer a holistic understanding of the classifier's efficacy, the performance metrics are delineated for both training and testing phases, facilitating a nuanced assessment of its capabilities across familiar and novel scenarios.

7.1. Experiment One—Pre-Trained CNN Models

In Experiment One, where ResNet50, MobileNet, and DenseNet121 underwent fine-tuning on the Dhad and Hijja datasets, several intriguing training dynamics were observed (see Figures 4 and 5). The training accuracy curves consistently exhibited a positive exponential trajectory, reflecting the models' progressive refinement and learning. Concurrently, the training loss curves showcased a negative exponential pattern, suggesting a consistent reduction in training errors—both patterns emblematic of typical training behaviour. While a nuanced performance advantage was discerned in favor of ResNet50 from the training curves, this superiority was marginal. However, the validation phase painted a slightly different picture. Although the validation curves initially mirrored the training trajectories, a noticeable degradation in performance became evident after a certain epoch. This divergence, particularly conspicuous in the validation loss curves post-epoch 7, is a clear manifestation of overfitting—a phenomenon exacerbated by the datasets' inherent simplicity. Such insights underscore the importance of leveraging validation metrics as they provide a clearer lens into the model's generalization prowess. Intriguingly, when evaluating based on validation performance, DenseNet121 emerged marginally superior, although

the performance disparities among the three models remained modest, positioning them comparably in terms of efficacy on these datasets.

A nuanced perspective on the models' training outcomes becomes apparent in the comparative analysis derived from Table 2. MobileNet emerged as the frontrunner for the Dhad dataset, boasting a validation loss of 0.2278 and a commendable accuracy of 0.9396. Conversely, for the Hijja dataset, DenseNet121 showcased its prowess with metrics of 0.4359 for validation loss and an accuracy score of 0.8920.

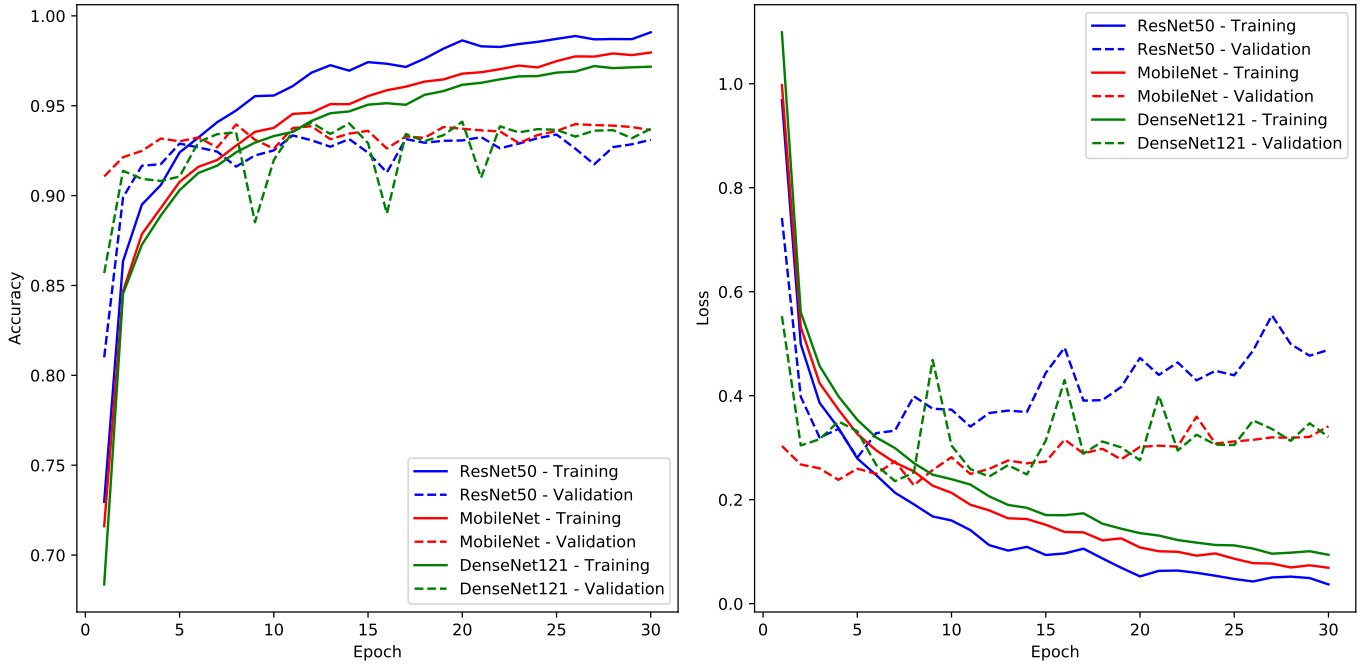


Figure 4. Training accuracy and loss curves for pre-trained models on Dhad dataset.

Delving deeper into the model performances, both MobileNet and DenseNet121 exhibited closely matched capabilities, with only marginal differences in their efficacy. In stark contrast, ResNet50's performance trajectory leaned more towards pruning, hinting at potential redundancy or inefficiencies in its architecture. This behaviour can be attributed to ResNet50's heavier design, which might have rendered it more susceptible to overfitting, especially given the datasets' inherent simplicity. In contrast, MobileNet's leaner architecture seemingly conferred upon it a more adaptive and resilient nature, enabling it to outperform its counterparts.

A comparative examination between the Dhad and Hijja datasets further illuminates this discussion. Predominantly, the pre-trained models showcased superior performance metrics on the Dhad dataset, underscoring its superior quality and efficacy in facilitating model training. Such observations align with the hypothesis positing Dhad's utilization of enhanced preprocessing methodologies, likely resulting in a cleaner, noise-attenuated dataset conducive for effective model learning. This superior data quality inherently empowered the models, enabling them to achieve heightened accuracies and reduced losses on the Dhad dataset compared to its Hijja counterpart.

The testing phase (see Table 3) further substantiated the models' capabilities, revealing outcomes that closely mirrored their validation performance. Such consistency underscores the models' adeptness at capturing generalized features, enabling them to maintain consistent performance across previously unseen datasets. Specifically, for the Dhad dataset, MobileNet continued to demonstrate its efficacy, registering a test accuracy of 0.9359, a test loss of 0.2468, and an impressive $F1$ score of 0.94. On the other hand, for the Hijja dataset, DenseNet121 emerged as the optimal performer, achieving a test accuracy of 0.8883, a test loss of 0.4919, and an $F1$-score of 0.89.

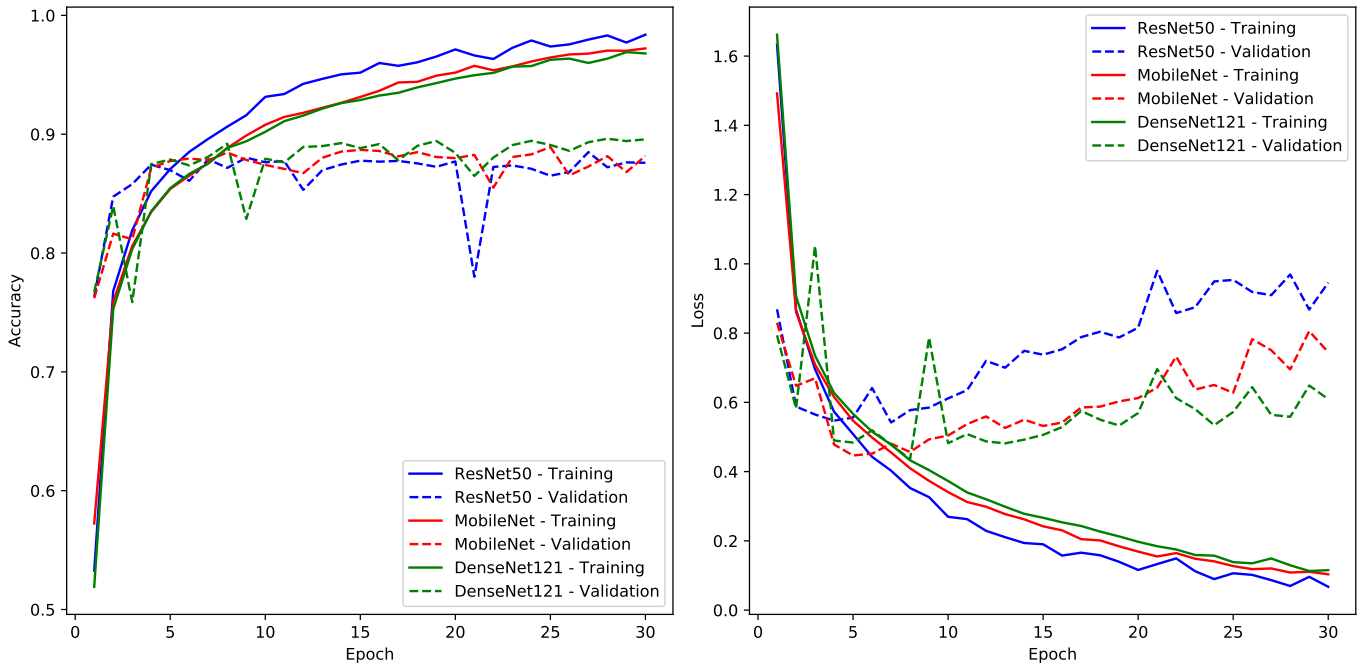


Figure 5. Training accuracy and loss curves for pre-trained models on Hijja dataset.

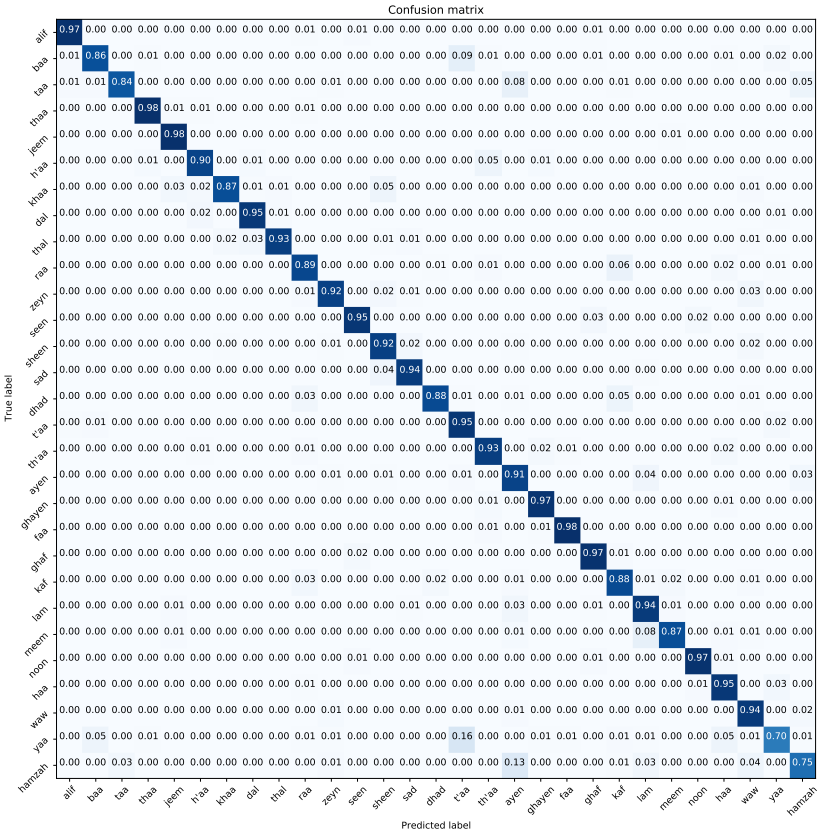
Table 2. Training performance of pre-trained models on Dhad and Hijja datasets.

	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
		Dhad Dataset		
MobileNet	0.0691	0.9796	0.2278	0.9396
DenseNet121	0.0941	0.9717	0.2357	0.9342
ResNet50	0.0371	0.9908	0.2810	0.9289
		Hijja Dataset		
MobileNet	0.1035	0.9722	0.4466	0.8774
DenseNet121	0.1154	0.9680	0.4359	0.8920
ResNet50	0.0674	0.9838	0.5419	0.8789

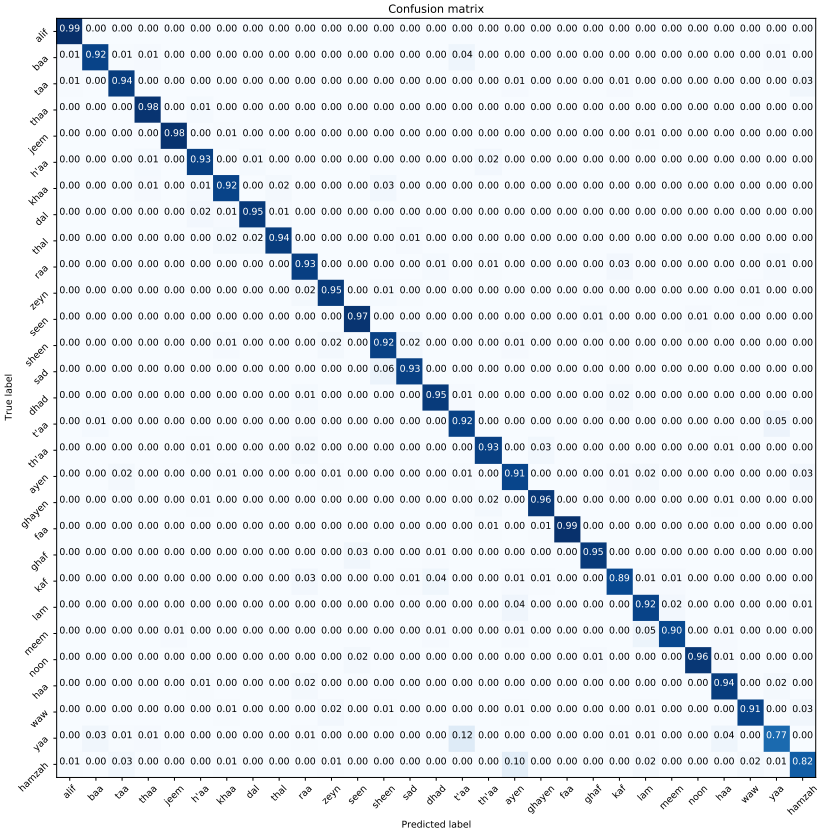
Figures 6 and 7 present the confusion matrices for the trained models on the Dhad and Hijja datasets, respectively. These matrices serve as pivotal tools for gauging the models' class-specific performances and identifying potential areas of misclassification.

Table 3. Test performance of pre-trained models on Dhad and Hijja datasets.

	Test Accuracy	Test Loss	*F*1 Score	Precision	Recall	J-Index
		Dhad Dataset				
MobileNet	0.9359	0.2468	0.94	0.94	0.94	0.88
DenseNet121	0.9306	0.2510	0.93	0.93	0.93	0.87
ResNet50	0.9228	0.3043	0.92	0.92	0.92	0.86
		Hijja Dataset				
MobileNet	0.8781	0.4677	0.88	0.88	0.88	0.78
DenseNet121	0.8883	0.4619	0.89	0.89	0.89	0.80
ResNet50	0.8705	0.6026	0.87	0.87	0.87	0.77

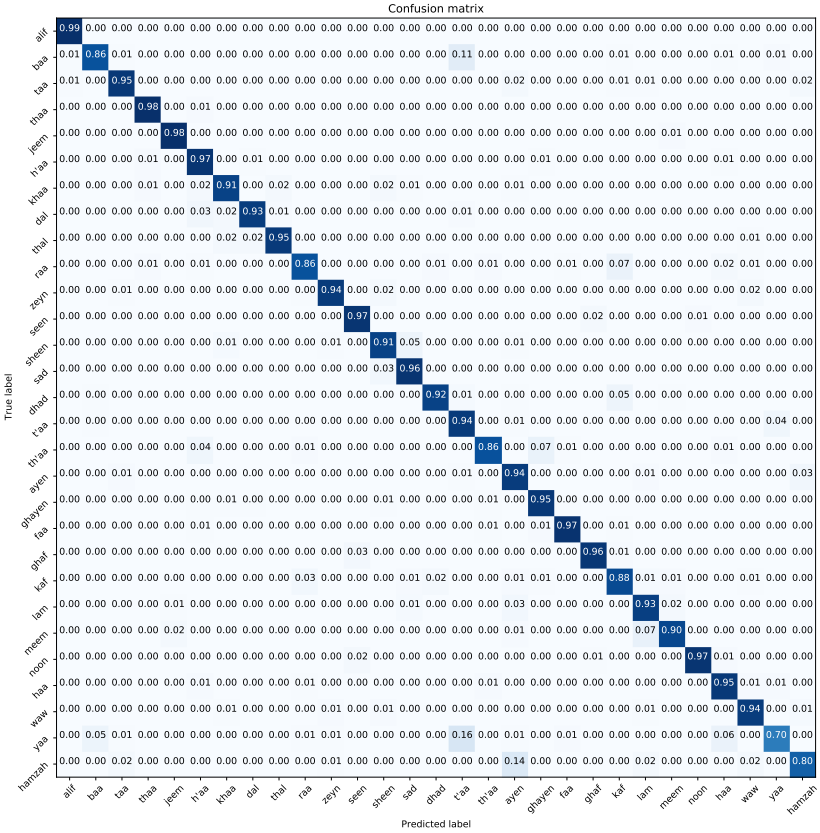


(**a**)



(**b**)

Figure 6. *Cont.*



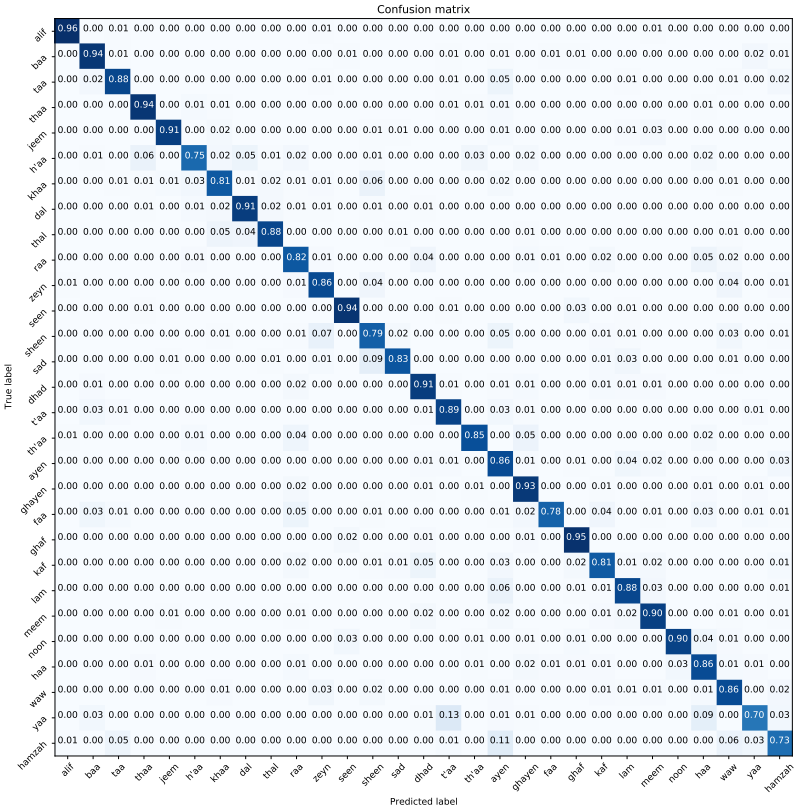
(**c**)

Figure 6. Confusion matrix for pre-trained models on Dhad dataset. (**a**) ResNet50, (**b**) MobileNet, and (**c**) DenseNet121.

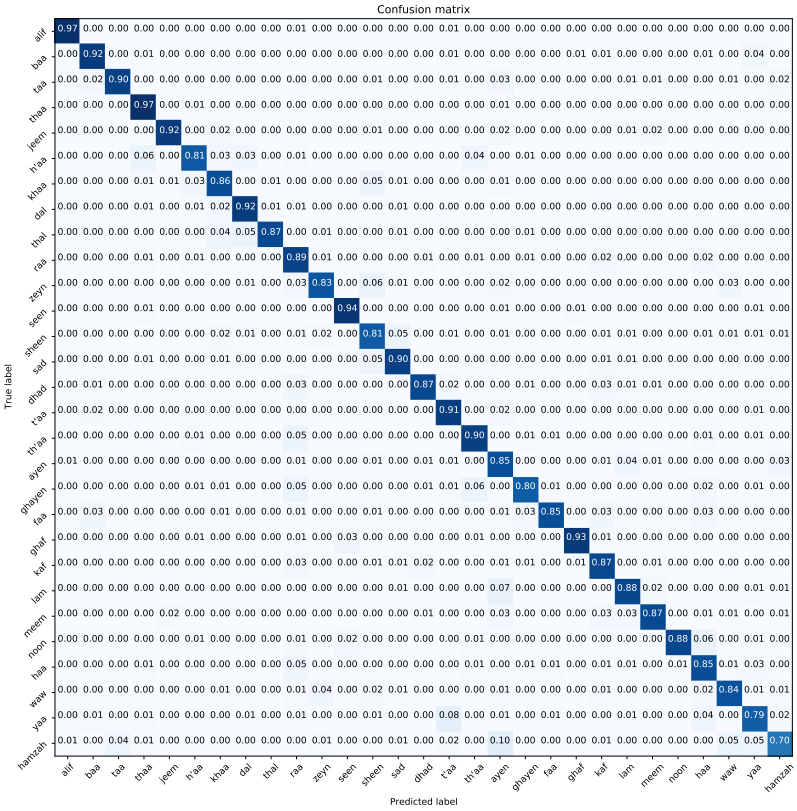
In the context of the Dhad dataset, a detailed examination reveals MobileNet's commendable class-wise equilibrium, characterized by minimal misclassifications across various categories. Notably, there's a discernible pattern of misclassification, where 12% of the "yaa" samples are erroneously categorized as "t'aa" and 10% of the "hamzah" samples are mislabeled as "ayen". Such misclassifications likely stem from the intricate visual similarities inherent to these characters, underscoring the inherent challenges of handwritten character recognition tasks.

Turning our attention to the Hijja dataset, DenseNet121 emerges as the model with the most consistent overall performance. However, a deeper dive into the confusion matrix reveals a higher incidence of misclassifications. Two salient observations include the misclassification of 10% of "lam" samples as "ayen" and 7% of "hamzah" instances being inaccurately labeled as "waw". Such misclassifications further emphasize the intricacies and challenges posed by handwritten Arabic character recognition, necessitating continuous refinement and optimization strategies for enhanced accuracy.

Figure 8 provides an insightful glimpse into the inner workings of the trained models through layer visualizations, shedding light on their training efficacy and decision-making processes. The two visualization techniques employed, Grad-CAM and SmoothGrad, serve distinct purposes in elucidating model behaviour. While Grad-CAM accentuates the pivotal regions within images that significantly influenced predictions, SmoothGrad offers a more granular perspective by pinpointing the specific pixels most instrumental in the decision-making process.

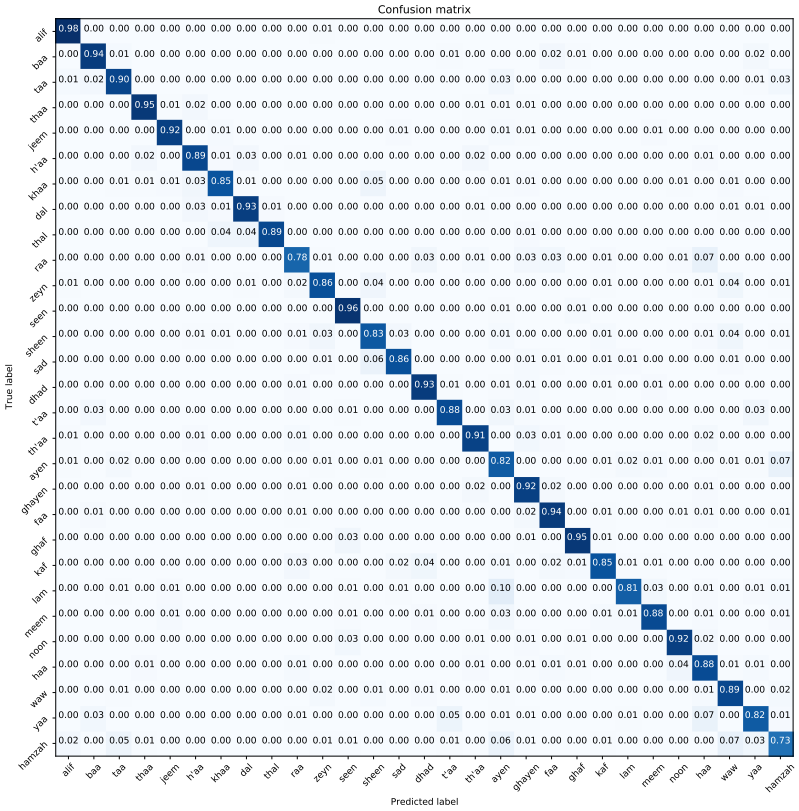


(**a**)



(**b**)

Figure 7. *Cont*.



(**c**)

Figure 7. Confusion matrix for pre-trained models on Hijja dataset. (**a**) ResNet50, (**b**) MobileNet, (**c**) DenseNet121.

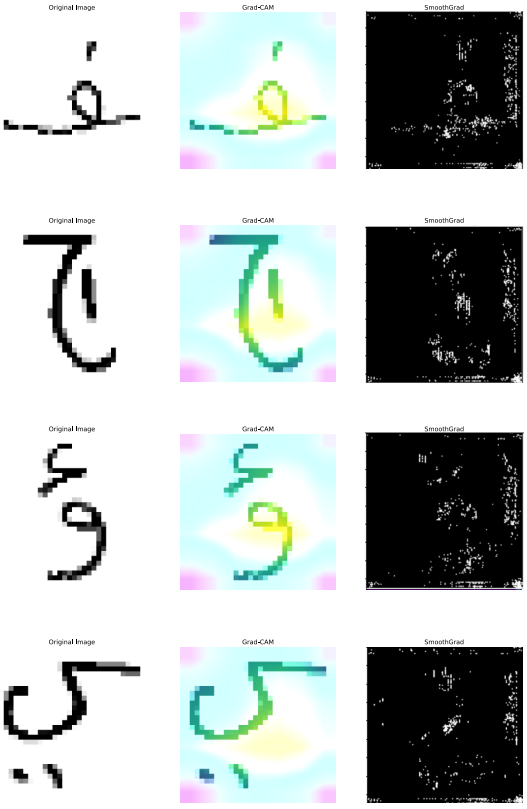


Figure 8. Layer visualizations for DenseNet121 model on Dhad dataset samples.

Upon meticulous examination of the visualizations, certain patterns and discrepancies come to the fore. Notably, for characters such as "faa" and "jeem", the models appear adept at capturing and leveraging the character-relevant pixels, indicative of robust training and feature extraction capabilities. However, a discernible shortfall becomes evident in the case of the "yaa" character. Here, the model seemingly overlooks or inadequately emphasizes crucial pixels during the prediction phase, suggesting potential areas for model refinement or additional training data augmentation to enhance accuracy and consistency.

Figures 9 and 10 present the AUC curves, offering a comprehensive overview of the discriminatory power and overall performance of the pre-trained models on the Dhad and Hijja datasets, respectively. AUC serves as a robust metric, encapsulating the model's ability to distinguish between different classes. Upon detailed examination of these curves, a pattern of closely matched performances across models emerges. Specifically, for the Dhad dataset, MobileNet slightly outperforms its counterparts, boasting an impressive AUC value of 0.9986. Conversely, on the Hijja dataset, DenseNet121 delivers a commendable performance, albeit marginally trailing behind MobileNet with an AUC of 0.9954.

From the experiments, it can be clearly observed that the models exhibited overfitting during training for both the Dhad and Hijja datasets for almost all the implemented models. Although the superior performance of the pre-trained models was recorded, it is important to take into consideration the overfitting problem. In general, this problem usually occurs when either the dataset is too small in comparison to the model complexity or the dataset is way too simple for the model. The literature suggests that dropout and data augmentation techniques can be used to overcome the overfitting problem. To further investigate this, in this experiment, we scoped the problem for only the MobileNet model on the Dhad dataset as a use case. We have tried different dropout ratios to observe the performance. Furthermore, we have also used data augmentation with the dropout. To be specific, we trained the model using the 0.2, 0.4, and 0.6 dropout values. In terms of data augmentation, we used rotation, width shift, height shift, shear, zoom, and nearest fill. Figures 11 and 12 show the trends for training and validation loss curves for both cases to understand. First, talking about the dropout variations, it can be observed from Figure 11 that a dropout percentage of 0.4 resulted in slightly better performance and a stable validation loss curve, whereas the dropout of 0.2 and 0.6 percentages degraded the performance. This suggests that not all the dropout percentages result in better performance; rather, an optimal value needs to be identified. It can be concluded that as suggested by the literature, dropout can be introduced to improve overfitting. In regard to the data augmentation and dropout variations, it can be observed from Figure 12 that the introduction of data augmentation did improve the overall training performance and resulted in emergence at lower loss values, but it did not really address the overfitting problem. However, when data augmentation was used with optimal dropout values i.e., 0.4, it resulted in a stable and improved validation loss curve. As a summary of this investigation, it can be concluded that overfitting is very common for smaller and simpler datasets. Dropout and data augmentation approaches can be used to improve overfitting to some extent; however, on a larger scale, the dataset needs to be introduced with noise and challenges to avoid this problem.

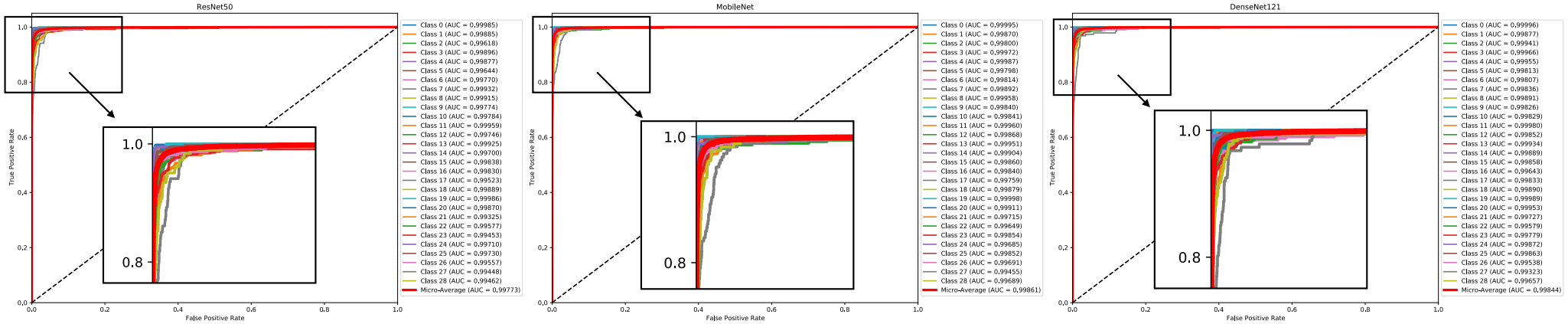


Figure 9. AUC curves for pre-trained models on Dhad dataset.

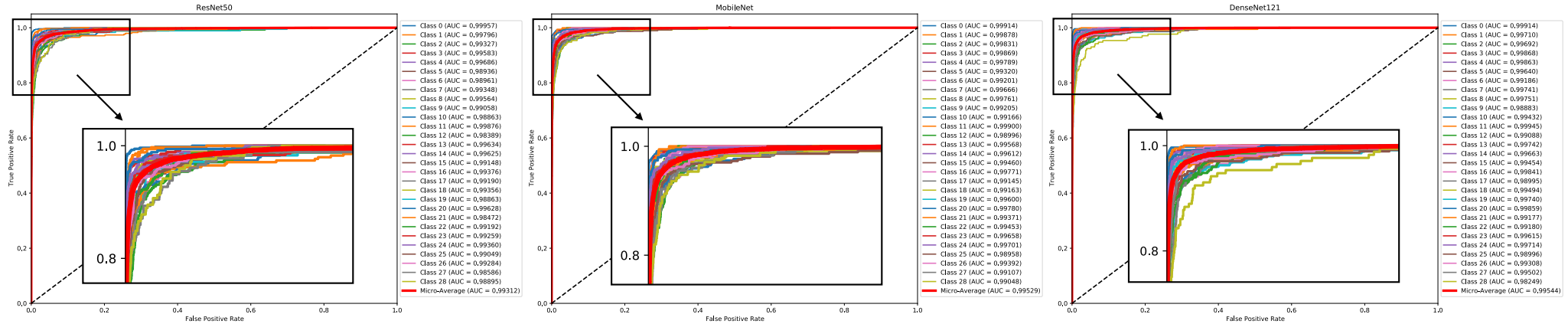


Figure 10. AUC curves for pre-trained models on Hijja dataset.

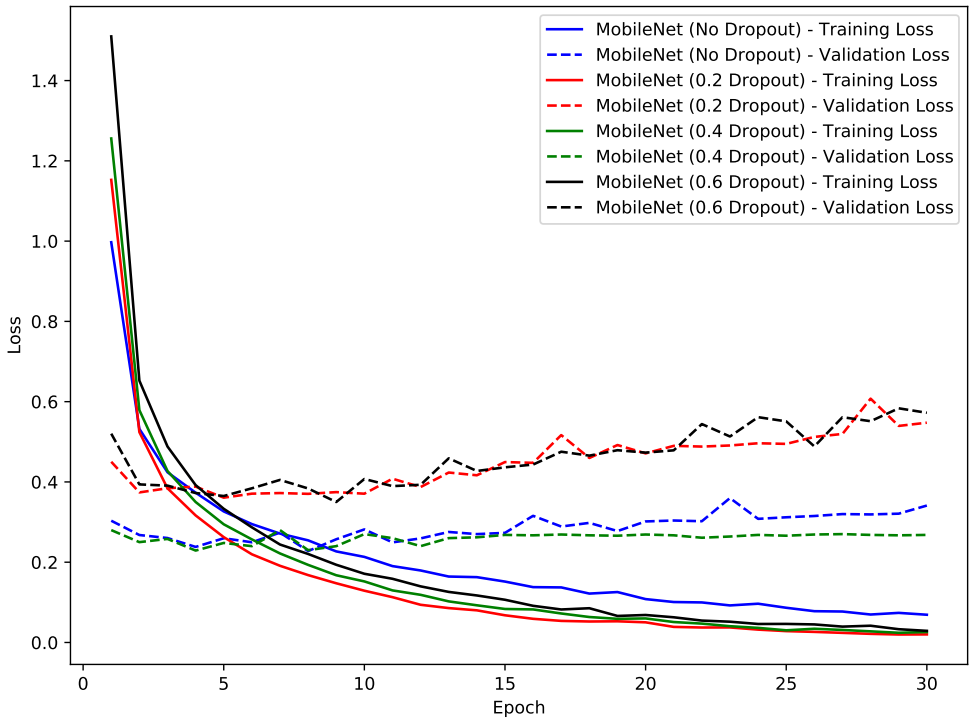


Figure 11. Loss curves for different dropout variations with MobileNet on Dhad dataset.

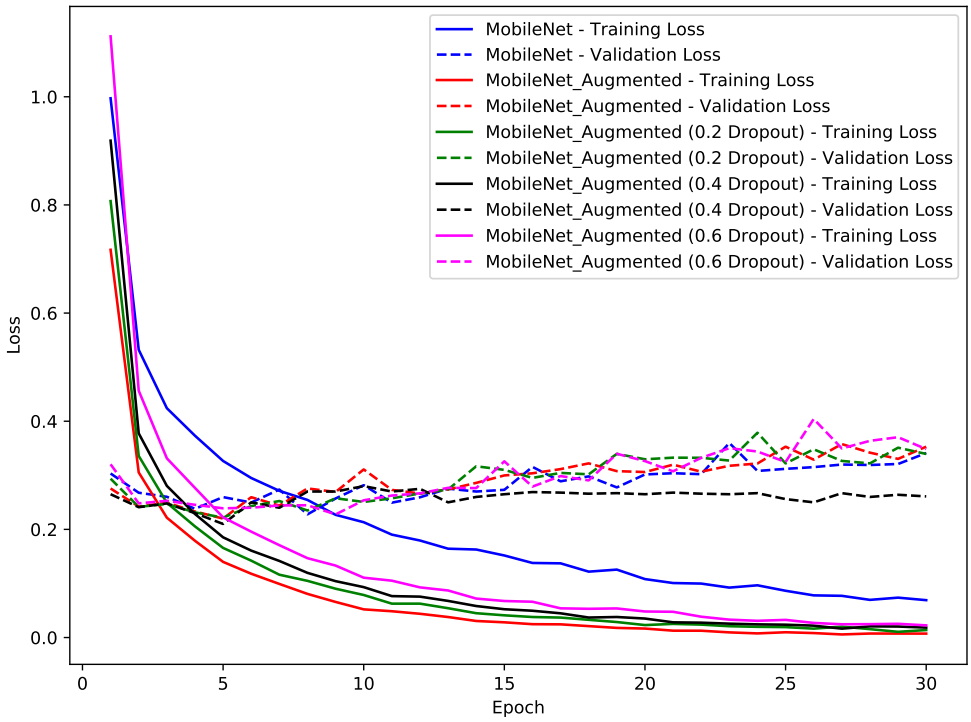


Figure 12. Loss curves for different dropout variations and data augmentation with MobileNet on Dhad dataset.

7.2. Experiment Two—Custom CNN Model

In Experiment Two, a custom CNN model, drawing inspiration from the existing literature [4,5,13–15], was meticulously crafted and subsequently trained on both the Dhad and Hijja datasets. A detailed analysis of the model's training dynamics, as depicted in Figure 13, offers invaluable insights into its performance and adaptability.

Upon scrutinizing the training accuracy plots, a discernible positive exponential trend is evident, aligning with typical training behaviour. However, an intriguing observation is the model's accelerated convergence, achieving desirable accuracy at a slightly quicker pace compared to other models. Nevertheless, the validation phase unraveled some concerns. While the validation accuracy initially mirrored the training trajectory, a conspicuous divergence emerged after the 20th epoch, signaling the onset of overfitting.

This overfitting propensity is further accentuated in the loss curves. After the 20th epoch, a palpable uptick in validation loss becomes evident, corroborating the overfitting suspicions. Such concerns are further compounded upon examining the convergence metrics; as detailed in Table 4, the model's loss values upon convergence are unexpectedly elevated for both the Dhad and Hijja datasets. Specifically, for the Dhad dataset, the model managed to attain a validation accuracy of 89% but manifested a relatively elevated validation loss of 0.3862. Conversely, the Hijja dataset witnessed a more pronounced performance disparity, with the model registering a diminished accuracy of 75% accompanied by a markedly higher validation loss of 0.8382.

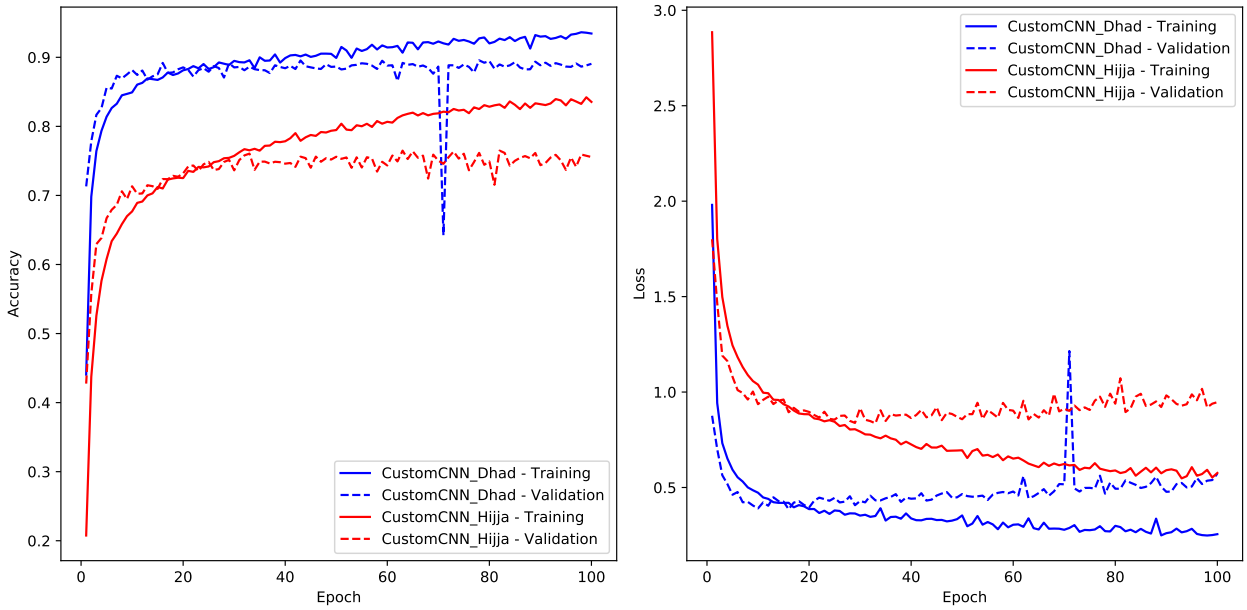


Figure 13. Training accuracy and loss curves for custom CNN model on Dhad and Hijja datasets.

Table 4. Training performance of custom CNN model on Dhad and Hijja datasets.

	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
Dhad Dataset	0.2554	0.9344	0.3862	0.8919
Hijja Dataset	0.5761	0.8354	0.8382	0.7515

Table 5 provides a comprehensive overview of the custom model's test performance metrics on both the Dhad and Hijja datasets. A cursory examination of these results reveals a coherent alignment with the model's validation trajectory, reaffirming the standard train–validate–test paradigm, where the performances across validation and test phases remain largely congruent.

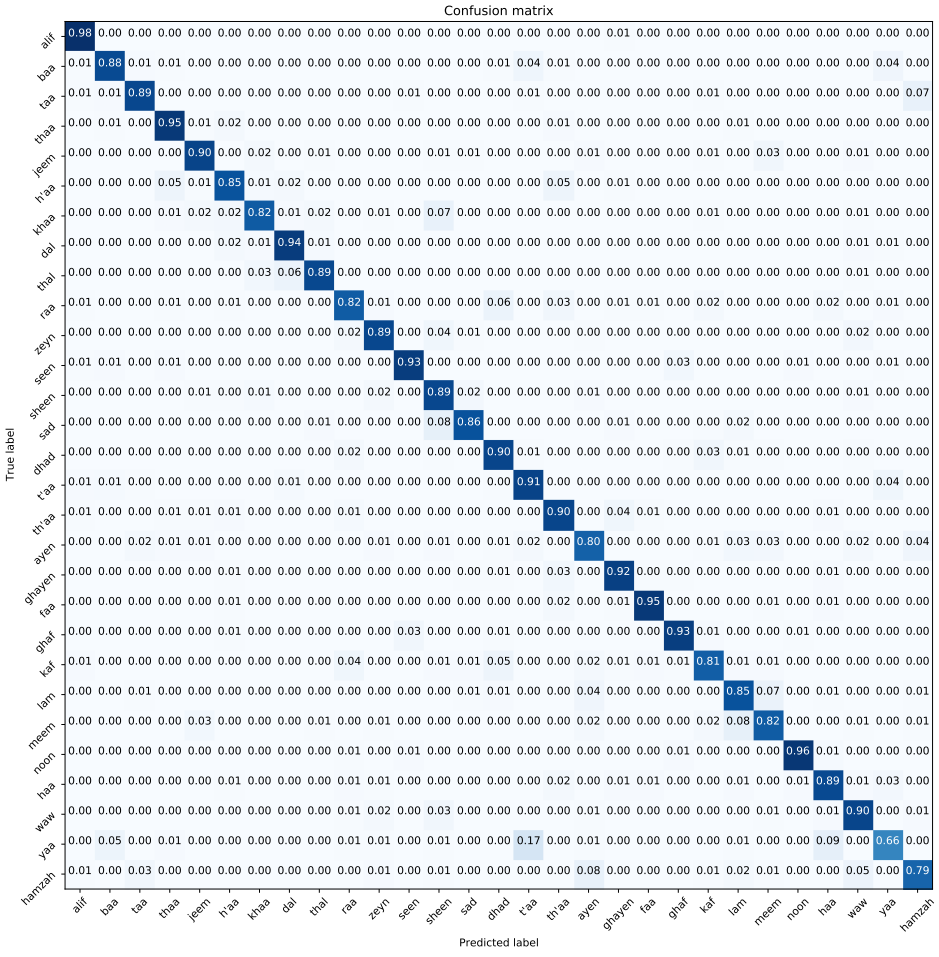
For the Dhad dataset, the custom model demonstrated a commendable test accuracy of 88%, coupled with a test loss metric of 0.3988. Additionally, the model's $F1$ score stood impressively at 0.89, underscoring its proficiency in maintaining a harmonious balance between precision and recall. Conversely, when evaluated on the Hijja dataset, the model's performance exhibited a discernible decline, registering a test accuracy of 74%. The associated test loss and $F1$ score metrics further elucidate this observation, standing at 0.8693 and 0.75, respectively.

Table 5. Test performance of custom CNN model for Dhad and Hijja datasets.

	Test Accuracy	Test Loss	*F*1 Score	Precision	Recall	J-Index
Dhad Dataset	0.8854	0.3988	0.89	0.89	0.89	0.79
Hijja Dataset	0.7484	0.8693	0.75	0.75	0.75	0.6

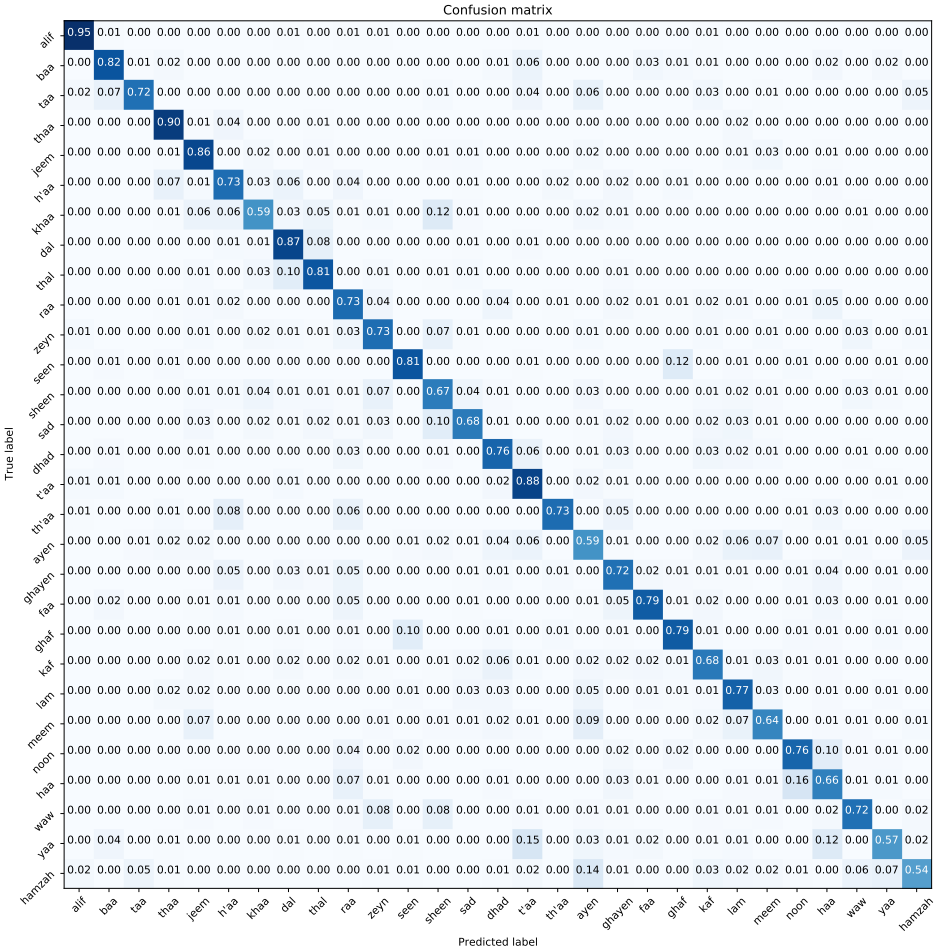
The nuanced performance trajectories across the two datasets are further corroborated by the AUC curves, meticulously depicted in Figure 14. The Dhad dataset witnessed a marginally superior performance, with the model achieving an AUC of 0.99, indicative of its robust discriminatory prowess. In stark contrast, the Hijja dataset, although exhibiting a commendable AUC value of 0.98, revealed a more scattered performance distribution across classes, emphasizing the inherent challenges and intricacies associated with character recognition tasks on this dataset.

Figure 15 provides a comprehensive confusion matrix for both the Dhad and Hijja datasets. While the model's performance on the Dhad dataset appears balanced with minimal misclassifications, a discernible decline is evident on the Hijja dataset, characterized by widespread misclassifications across various classes. Particularly challenging are the class pairs "t'aa–yaa" and "ayen–hamza", likely due to their visual resemblance, underscoring the inherent complexities in Arabic character recognition and highlighting areas for potential model enhancement.



(**a**)

Figure 14. *Cont.*



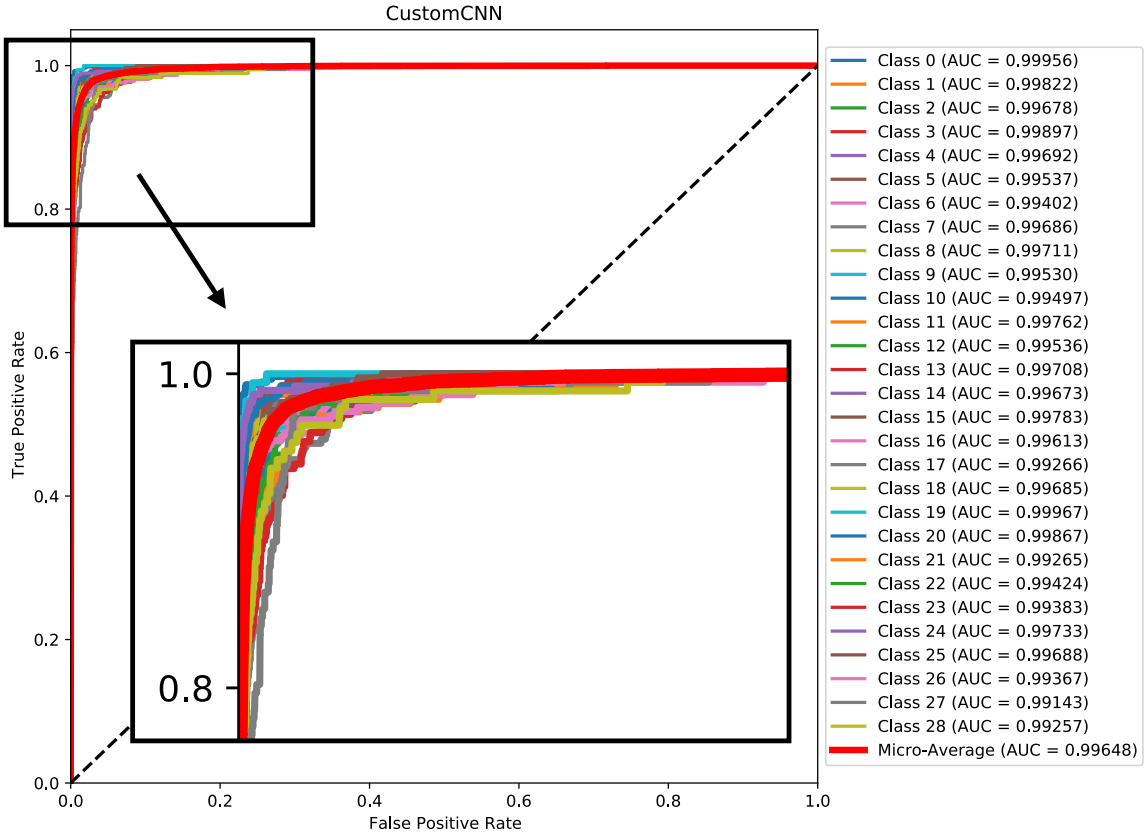
(**b**)

Figure 14. Confusion matrix for custom CNN model on Dhad and Hijja datasets. (**a**) Dhad and (**b**) Hijja.

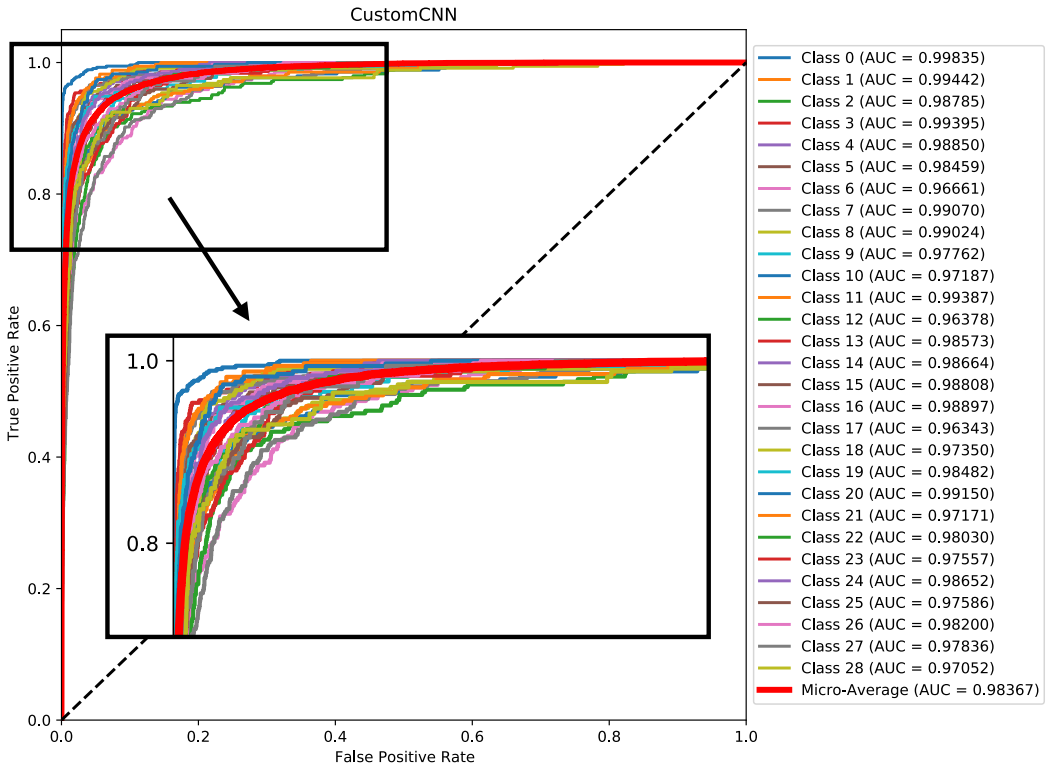
7.3. Experiment Three—Classification of Deep Visual Features

In Experiment Three, a sophisticated two-stage approach was devised to optimize the classification process, blending the strengths of deep learning feature extraction with the precision of traditional classifiers. The foundational component of this pipeline was the MobileNet architecture, renowned for its prowess in extracting intricate features from complex datasets. By utilizing MobileNet's capabilities, the experiment aimed to transform the raw data into a more discernible and compact representation, thereby facilitating more effective subsequent classification. In this context, we have used the MobileNet model pre-trained over the ImageNet and MobileNet models trained in Experiment One.

Following the feature extraction phase, the extracted features were then subjected to three distinct conventional classifiers: SVM, RF, and MLP. SVM, a discriminative classifier, operates by finding the optimal hyperplane that best separates the data into distinct classes, making it particularly adept at handling high-dimensional feature spaces. Conversely, RF, an ensemble learning method, constructs multiple decision trees during training and outputs the class that is the mode of the classes of individual trees for classification tasks, thereby leveraging the wisdom of multiple trees to enhance accuracy and robustness. On the other hand, MLP is known for its fully connected neural architecture to extract the hidden patterns from the input feature vector.



(**a**)



(**b**)

Figure 15. AUC curves of custom CNN model for Dhad and Hijja datasets. (**a**) Dhad and (**b**) Hijja.

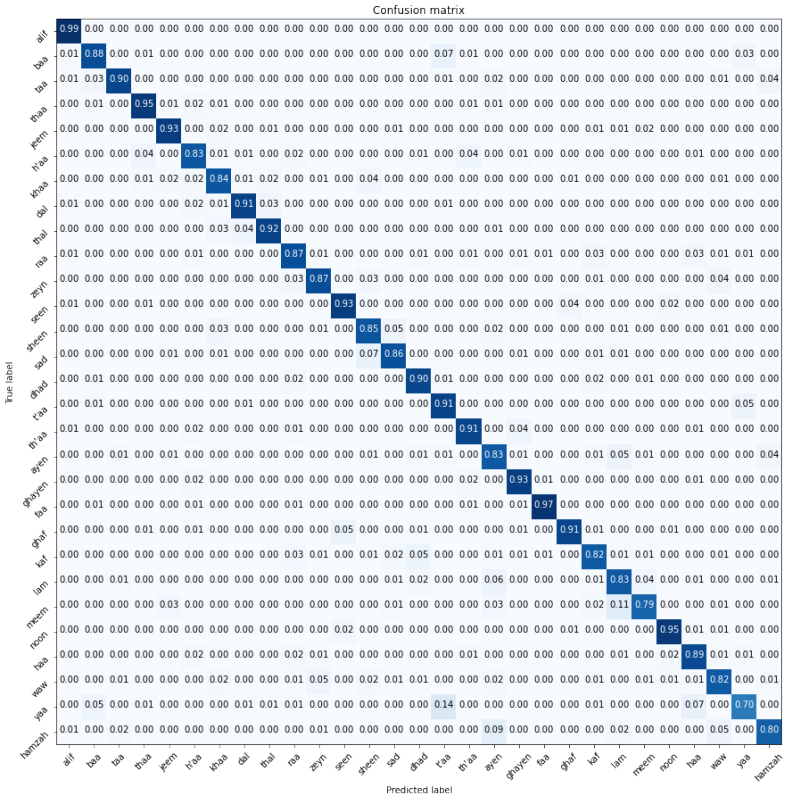
Delving into the results encapsulated in Table 6, a discernible pattern emerges. For the Dhad dataset, the MobileNet + SVM ensemble manifested as the optimal configuration, demonstrating its prowess with a validation accuracy of 89%, which was corroborated by the test accuracy standing at a commendable 88%. Further fortifying its performance credentials, the ensemble yielded an *F*1 score of 0.88, underscoring its balanced precision and recall capabilities. Similarly, when transposed to the Hijja dataset, the MobileNet+SVM configuration continued its dominance, albeit with slightly diminished metrics. A validation accuracy of 73% and a corresponding test accuracy of 72% were achieved, along with an *F*1 score of 0.73, signifying a robust performance despite the dataset's inherent complexities. In context to the use of ImageNet pre-trained and Experiment One trained model, it can be observed that the ImageNet pre-trained model resulted in better performance.

Table 6. Test performance of two-stage classification on deep visual features pipeline on Dhad and Hijja datasets.

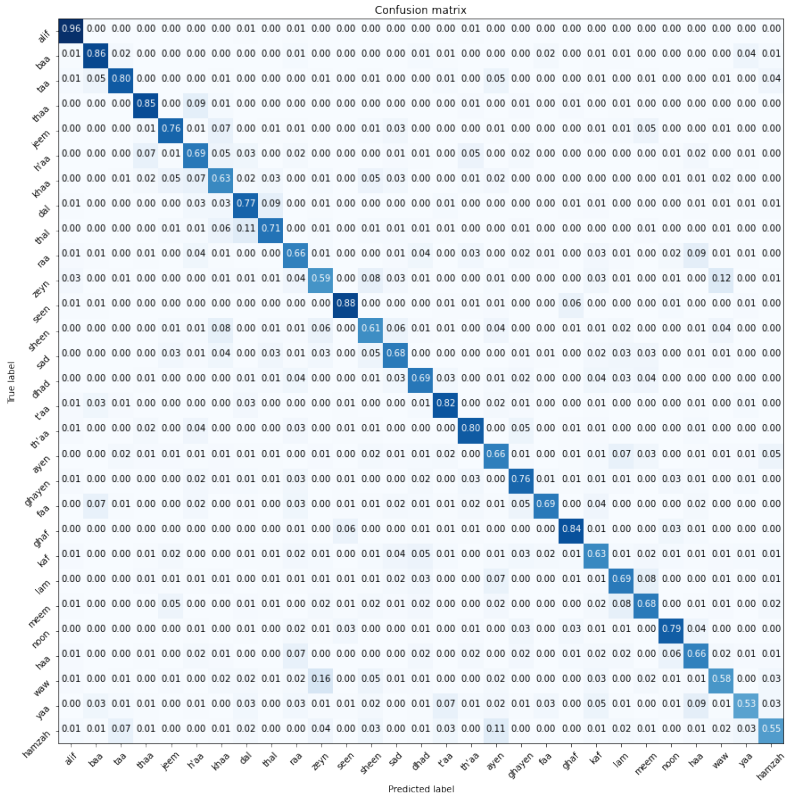
	Validation Accuracy	**Test Accuracy**	***F*1 Score**	**Precision**	**Recall**	**J-Index**
Dhad Dataset						
MobileNet (ImageNet Pre-Trained) + SVM	0.8894	0.8848	0.88	0.88	0.88	0.79
MobileNet (ImageNet Pre-Trained) + RF	0.7775	0.7803	0.78	0.78	0.78	0.64
MobileNet (ImageNet Pre-Trained) + MLP	0.0801	0.0810	0.02	0.02	0.02	0.05
MobileNet (Experiment One) + SVM	0.7118	0.7100	0.71	0.71	0.71	0.71
MobileNet (Experiment One) + RF	0.6662	0.6656	0.66	0.65	0.66	0.50
MobileNet (Experiment One) + MLP	0.0556	0.0556	0.01	0.01	0.01	0.03
Hijja Dataset						
MobileNet (ImageNet Pre-Trained) + SVM	0.7322	0.7256	0.73	0.73	0.73	0.57
MobileNet (ImageNet Pre-Trained) + RF	0.5346	0.5270	0.53	0.52	0.53	0.36
MobileNet (ImageNet Pre-Trained) + MLP	0.0770	0.0771	0.04	0.04	0.04	0.06
MobileNet (Experiment One) + SVM	0.3937	0.3839	0.37	0.38	0.37	0.24
MobileNet (Experiment One) + RF	0.3705	0.3591	0.34	0.35	0.34	0.22
MobileNet (Experiment One) + MLP	0.0578	0.0570	0.01	0.01	0.03	0.03

Figures 16–21 provide a detailed representation of the confusion matrices derived from the SVM, RF, and MLP classifiers when employed with deep visual features from the ImageNet pre-trained and Experiment One trained MobileNet model on both the Dhad and Hijja datasets. Complementing these visual representations, the findings elucidated in the table corroborate the classifiers' performance metrics. Notably, SVM emerges as the superior performer across all the cases.

However, when contextualized within the datasets, a nuanced observation surfaces. The Dhad dataset consistently showcases enhanced performance metrics in comparison to its Hijja counterpart. This disparity in performance underscores the Dhad dataset's superior quality, likely attributed to meticulous data curation, reduced noise levels, or other preprocessing enhancements. Such insights are pivotal, as they not only validate the efficacy of the classification pipeline but also emphasize the pivotal role of dataset quality in influencing model performance and outcomes.

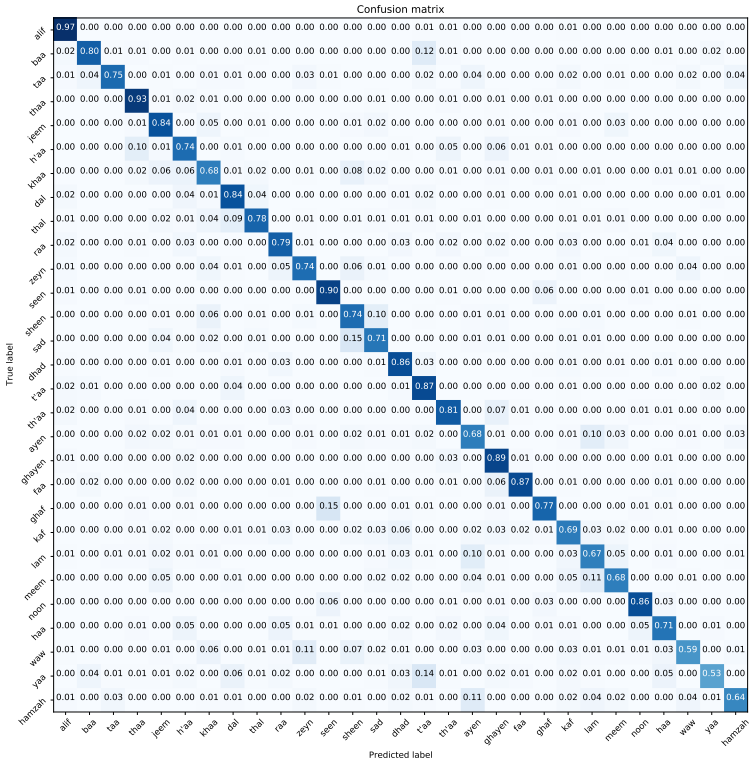


(**a**)

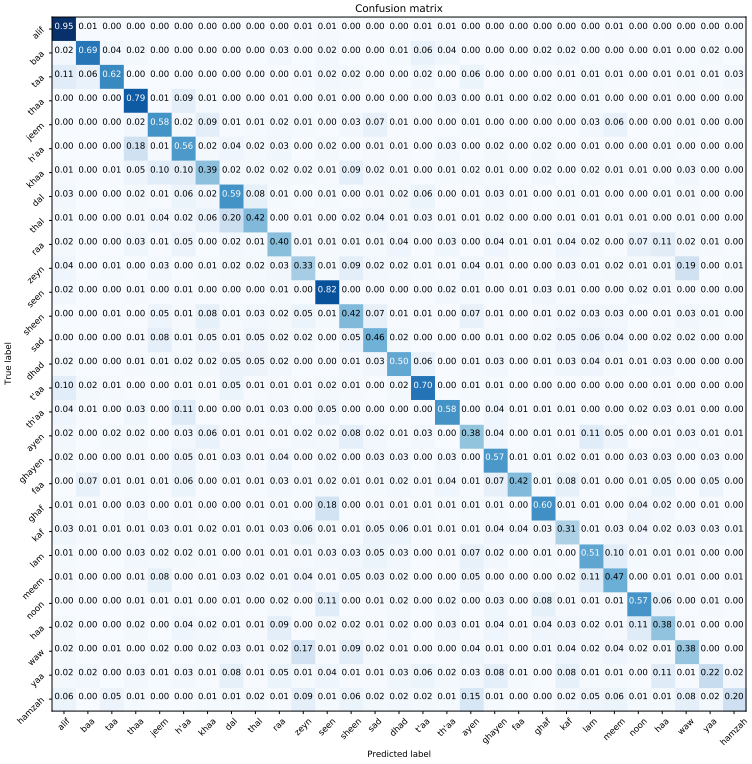


(**b**)

Figure 16. Confusion matrix of MobileNet (ImageNet pre-trained) + SVM pipeline on Dhad and Hijja datasets. (**a**) Dhad and (**b**) Hijja.

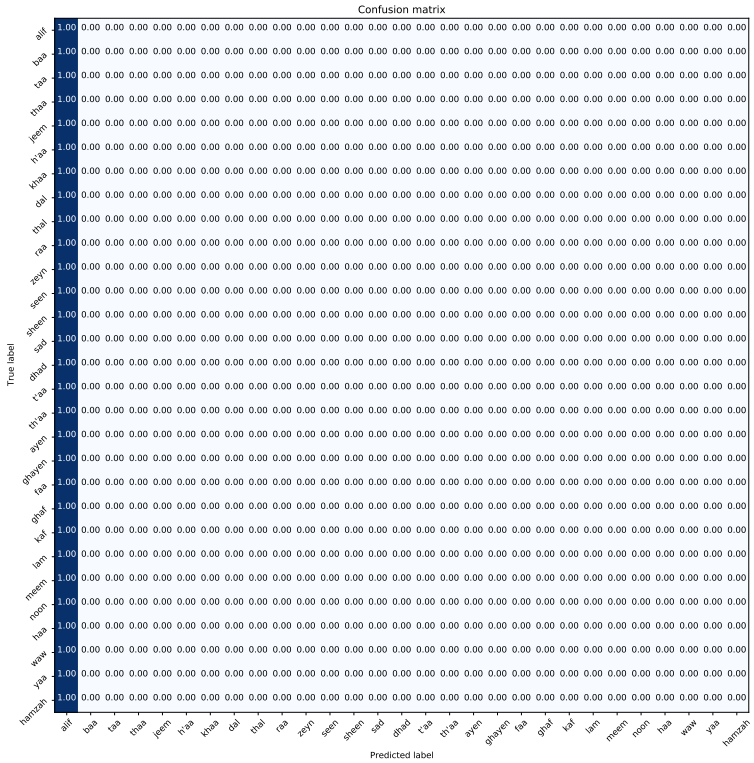


(**a**)

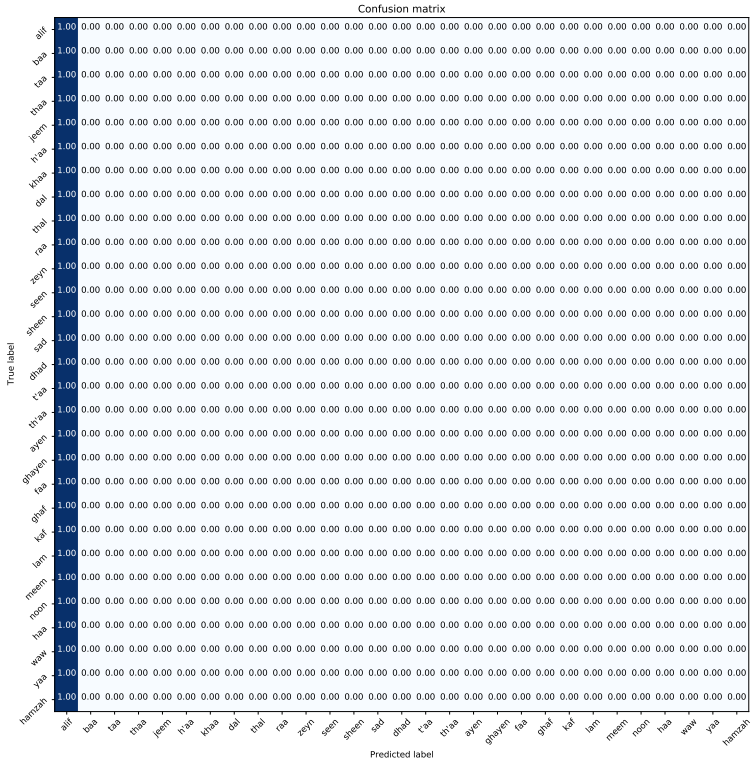


(**b**)

Figure 17. Confusion matrix of MobileNet (ImageNet pre-trained) + RF pipeline on Dhad and Hijja datasets. (**a**) Dhad and (**b**) Hijja.

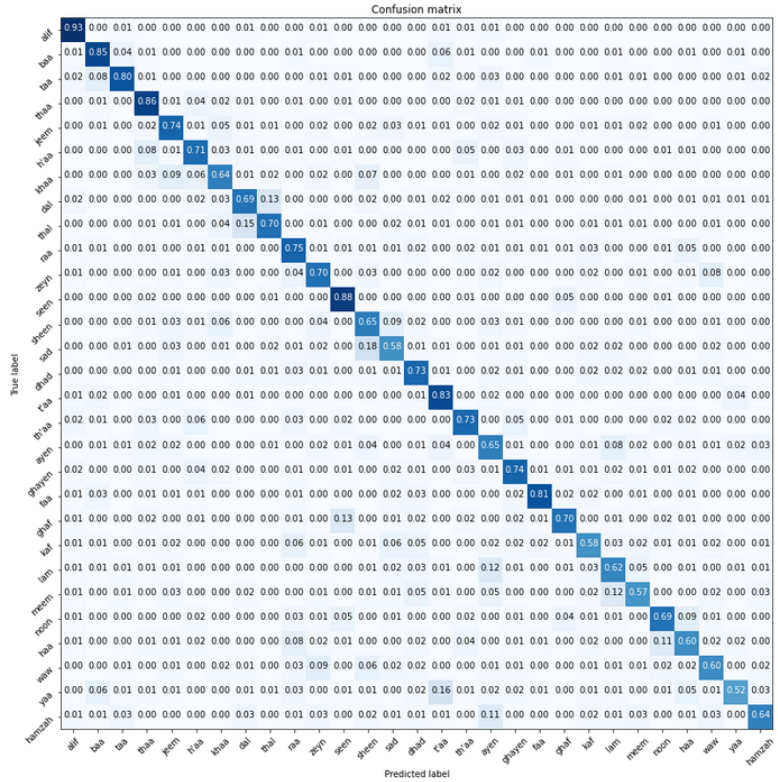


(**a**)

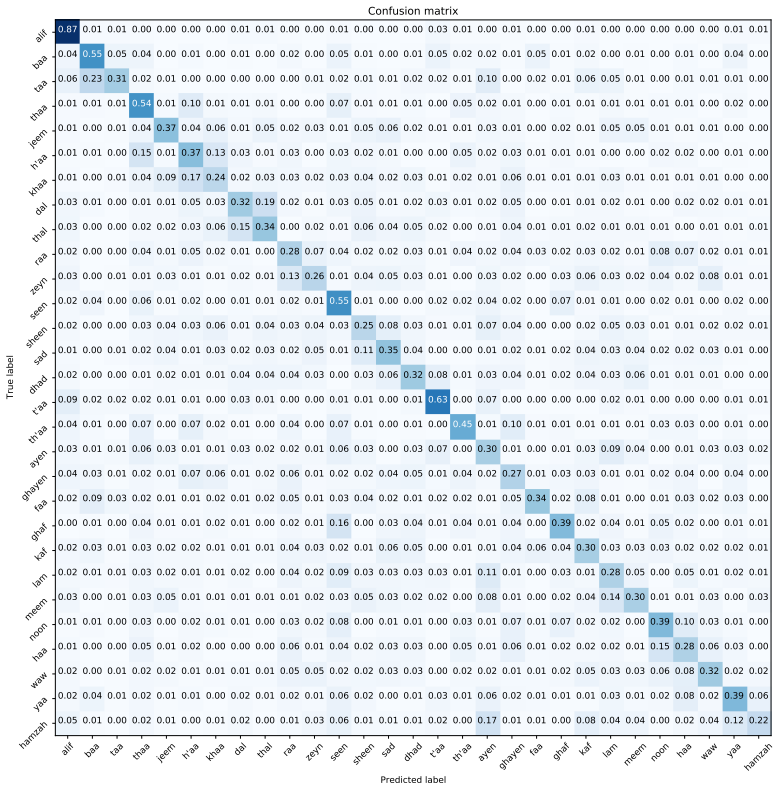


(**b**)

Figure 18. Confusion matrix of MobileNet (ImageNet pre-trained) + MLP pipeline on Dhad and Hijja datasets. (**a**) Dhad and (**b**) Hijja.

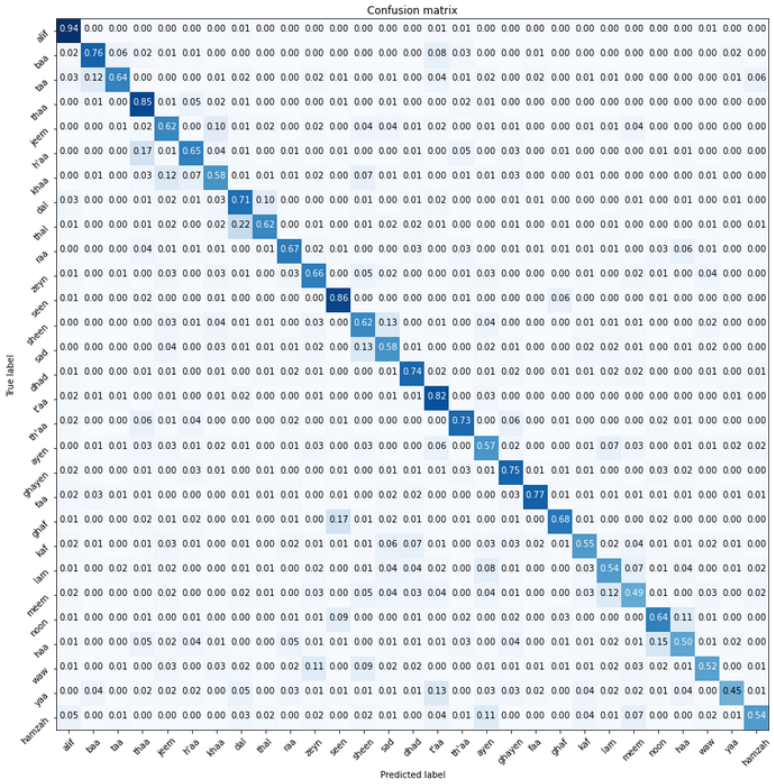


(**a**)

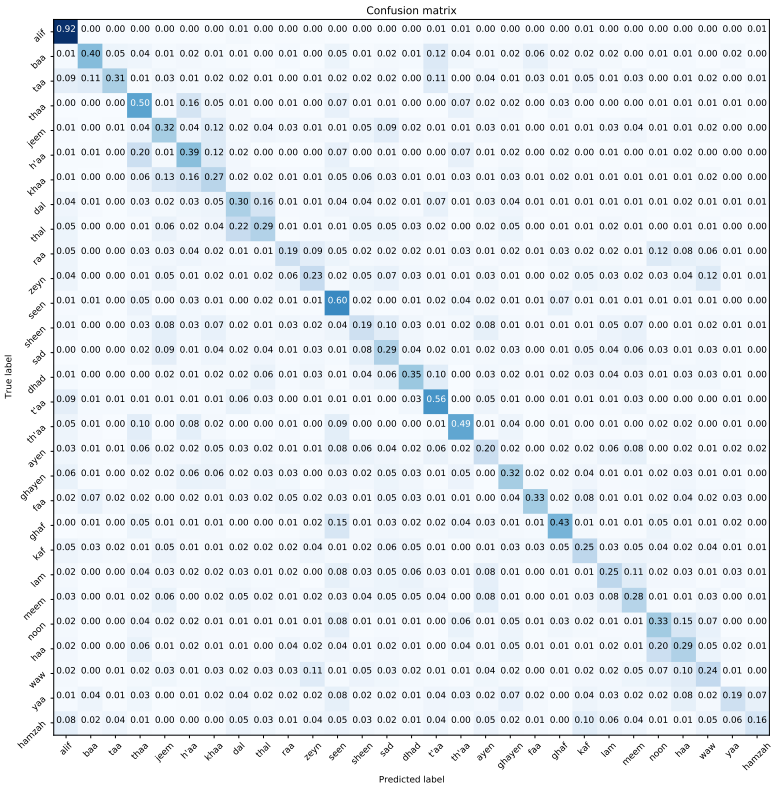


(**b**)

Figure 19. Confusion matrix of MobileNet (Experiment One) + SVM pipeline on Dhad and Hijja datasets. (**a**) Dhad and (**b**) Hijja.

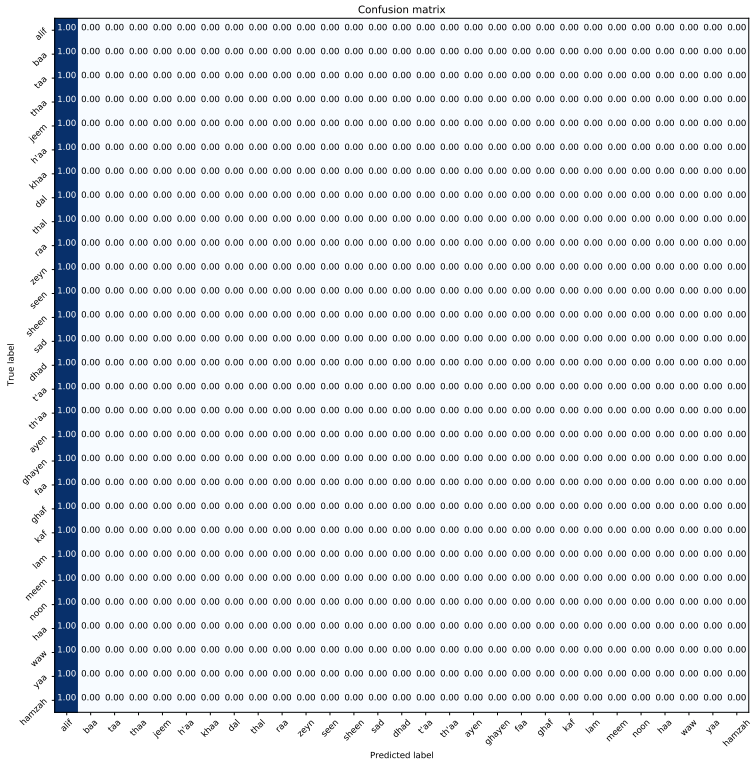


(**a**)

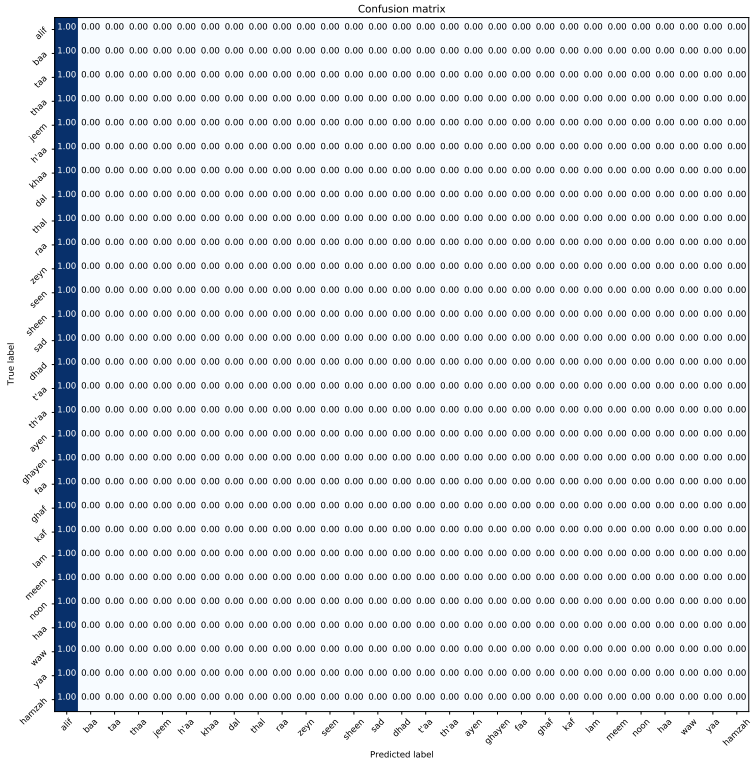


(**b**)

Figure 20. Confusion matrix of MobileNet (Experiment One) + RF pipeline on Dhad and Hijja datasets. (**a**) Dhad and (**b**) Hijja.



(**a**)



(**b**)

Figure 21. Confusion matrix of MobileNet (Experiment One) + MLP pipeline on Dhad and Hijja datasets. (**a**) Dhad and (**b**) Hijja.

8. Discussion

The exploration of handwritten Arabic character recognition, particularly among children, presents both challenges and opportunities. The results obtained from the experiments provided valuable insights into the effectiveness of deep learning models and traditional classifiers for this specific task. In this discussion, we delve deeper into the insights derived from the results, critically analysing them against the existing literature, evaluating the methodologies employed and highlighting potential avenues for future research.

- Fine-tuning of Existing Models: An important insight from the performed experiments is the unparalleled efficacy of fine-tuning existing deep learning architectures. This approach resonates with the existing literature [29–31], highlighting the potential of harnessing pre-trained models fine-tuned for application-specific tasks. The flexibility of fine-tuning, which combines using general features with adjusting to specific dataset details, highlights its essential importance. Especially in situations with limited computing power, its ability to achieve impressive results quickly becomes clearly noticeable.
- Custom CNN Model: In comparison to the established deep architectures, our simpler custom CNN achieved good results in classification. However, the performance did not exceed the fine-tuned pre-trained models. These findings align with what is commonly discussed in current research [32,33], emphasizing that simpler models can be easily affected by minor changes. This highlights the need for cautious interpretations and emphasizes the extra computing work needed when starting from scratch with new models.
- Two-stage Pipeline with Conventional Classifiers: Our exploration of a two-part process, combining deep visual feature extraction with traditional classification methods, resulted in less-than-ideal results. These results are consistent with existing research, highlighting the importance of end-to-end deep learning models trained effectively at once. The shortcomings arising from separate feature extraction and classification emphasize the need for unified model training, bringing together all elements to better achieve the main goal.
- Dataset Dynamics: At the heart of the model's performance variations lies the quality of the dataset. Our studies highlight the superiority of the Dhad dataset when compared to the Hijja one, likely due to clearer pixels and reduced interference. These insights highlight the crucial importance of careful dataset preparation, underscoring its fundamental role in shaping the best possible model results.
- Navigating Class Confounders: A recurring pattern throughout our experimental journey centres on the differentiation between certain class pairs, particularly "t'aa–yaa" and "ayen–hamzah". The blending of visual similarities among these classes leads to frequent misclassifications, highlighting the need for future efforts to develop more detailed training samples. Tackling this challenge requires a focused effort to enhance the dataset with diverse class examples, enhancing the model's ability to accurately distinguish categories.

While the experiments offer valuable insights, they are not devoid of limitations. The use of a limited number of datasets, potential biases in data curation, and the absence of real-world noise simulations may limit the external validity of the findings. Furthermore, the focus on specific architectures and classifiers suggests a comprehensive exploration of the deep learning and traditional machine learning models. Potential future research directions can be as follows:

1. Enhanced dataset curation, incorporating diverse writing styles, variations, and real-world noise simulations.
2. Comparative evaluations encompassing a broader spectrum of architectures, optimization techniques, and data augmentation strategies.
3. Exploration of ensemble methodologies, blending the strengths of multiple models to foster enhanced recognition capabilities.

9. Al-Khatta—An Early Intervention Tool for Arabic Handwriting Improvement

In envisioning a future application, a model specifically trained to classify Arabic characters holds immense potential for a highly impactful use-case "Al-Khatta" for the enhancement of Arabic handwriting skills in children aged 7 to 12. This software application can seamlessly integrate the trained model with the aim of revolutionizing handwriting improvement through innovative features. The model's real-time analysis capabilities will enable the application to deliver immediate feedback on handwritten input, fostering a dynamic and responsive learning environment. The trained model will play a pivotal role in identifying areas of difficulty within specific characters, empowering the app to generate personalized practice exercises tailored to each child's unique handwriting challenges. This forward-looking approach ensures a targeted and individualized learning experience, effectively addressing the diverse needs of young learners and fostering accelerated proficiency in Arabic handwriting.

Moreover, the application can incorporate a progress-tracking functionality, providing insightful data on a child's development across various exercises and over time. This feature will empower educators and parents with a comprehensive understanding of learning patterns, facilitating informed and targeted guidance to further support the child's progress. To maintain engagement in this envisioned future, the application can employ gamified elements and rewards, contributing to a positive reinforcement learning experience. By infusing an element of enjoyment into the learning process, the application aims to keep children motivated and enthusiastic about refining their Arabic handwriting skills.

The UI/UX development of the application can utilize HTML, CSS, and JavaScript for web-based applications or consider platform-specific frameworks such as Flutter for cross-platform mobile applications. In the realm of model development, PyTorch, TensorFlow, and Python libraries can be harnessed, with a dedicated GPU machine ensuring efficient training. For swift real-time performance in mobile deployment, models like MobileNet can be employed, while larger models like DenseNet121 may be considered for potential offline analysis.

10. Conclusions

In conclusion, our comprehensive exploration into the classification of handwritten Arabic characters among children reveals intriguing dynamics in model performance and dataset efficacy. While fine-tuned pre-existing models showcased commendable accuracy, particularly MobileNet on the Dhad dataset and DenseNet121 on the Hijja dataset, their performance trajectories underscored the challenges of overfitting, especially with datasets of inherent simplicity. The nuances observed in misclassifications, notably between visually similar characters, highlight the intricacies inherent to Arabic character recognition. A concept of computer application to facilitate the handwriting improvement in children is also discussed as a practical use-case of Arabic children's handwritten character recognition. Moving forward, addressing these challenges will demands multi-pronged approach: refining dataset quality, exploring advanced model architectures, and integrating robust training strategies to enhance generalization and accuracy.

Author Contributions: Conceptualization, N.A.; methodology, S.A. and N.A.; software, A.D.A., D.A., and R.A.; validation, H.A. and A.M.A.; data curation, D.A., R.A., H.A. and A.M.A.; writing—original draft preparation, S.A.; writing—review and editing, N.A.; visualization, A.D.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by King Saud University, Riyadh, Saudi Arabia, through Researchers Supporting Project number (RSPD2024R857).

Institutional Review Board Statement: This study was approved by the "Standing Committee on Ethics of Scientific Research" at King Saud University (No. KSUKSU-HE-19-363).

Data Availability Statement: The datasets generated and/or analysed during the current study are available in the Github repository, https://github.com/daadturki1/Dhad (accessed on 1 October 2023).

Acknowledgments: The authors thank the anonymous reviewers for their constructive comments.

Conflicts of Interest: The authors declare no competing interests.

References

1. Eberhard, D.M.; Simons, G.F.; Fennig, C.D. *Ethnologue: Languages of the World*; SIL International: Dallas, TX, USA 2023.
2. Nahar, K.M.; Alsmadi, I.; Al Mamlook, R.E.; Nasayreh, A.; Gharaibeh, H.; Almuflih, A.S.; Alasim, F. Recognition of Arabic Air-Written Letters: Machine Learning, Convolutional Neural Networks, and Optical Character Recognition (OCR) Techniques. *Sensors* **2023**, *23*, 9475. [CrossRef]
3. Kasem, M.S.; Mahmoud, M.; Kang, H.S. Advancements and Challenges in Arabic Optical Character Recognition: A Comprehensive Survey. *arXiv* **2023**, arXiv:2312.11812.
4. Altwaijry, N.; Al-Turaiki, I. Arabic handwriting recognition system using convolutional neural network. *Neural Comput. Appl.* **2021**, *33*, 2249–2261. [CrossRef]
5. Alwagdani, M.S.; Jaha, E.S. Deep Learning-Based Child Handwritten Arabic Character Recognition and Handwriting Discrimination. *Sensors* **2023**, *23*, 6774. [CrossRef]
6. Zakraoui, J.; Saleh, M.; Al-Maadeed, S.; AlJa'am, J.M. A study of children emotion and their performance while handwriting Arabic characters using a haptic device. *Educ. Inf. Technol.* **2023**, *28*, 1783–1808. [CrossRef]
7. El-Sawy, A.; Loey, M.; El-Bakry, H. Arabic handwritten characters recognition using convolutional neural network. *WSEAS Trans. Comput. Res.* **2017**, *5*, 11–19.
8. Lamghari, N.; Raghay, S. Recognition of Arabic Handwritten Diacritics using the new database DBAHD. In *Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2021; Volume 1743, p. 012023.
9. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
10. Fukushima, K.; Miyake, S. Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Visual Pattern Recognition. In *Competition and Cooperation in Neural Nets*; Amari, S.I., Arbib, M.A., Eds.; Springer: Berlin/Heidelberg, Germany, 1982; pp. 267–285.
11. Al-Turaiki, I.; Altwaijry, N. Hijja Dataset. 2019. Available online: https://github.com/israksu/Hijja2 (accessed on 10 January 2024).
12. Alkhateeb, J.H. An effective deep learning approach for improving off-line arabic handwritten character recognition. *Int. J. Softw. Eng. Comput. Syst.* **2020**, *6*, 53–61.
13. Nayef, B.H.; Abdullah, S.N.H.S.; Sulaiman, R.; Alyasseri, Z.A.A. Optimized leaky ReLU for handwritten Arabic character recognition using convolution neural networks. *Multimed. Tools Appl.* **2022**, *81*, 2065–2094. [CrossRef]
14. Alheraki, M.; Al-Matham, R.; Al-Khalifa, H. Handwritten Arabic Character Recognition for Children Writing Using Convolutional Neural Network and Stroke Identification. *Hum.-Centric Intell. Syst.* **2023**, *3*, 147–159. [CrossRef]
15. Bin Durayhim, A.; Al-Ajlan, A.; Al-Turaiki, I.; Altwaijry, N. Towards Accurate Children's Arabic Handwriting Recognition via Deep Learning. *Appl. Sci.* **2023**, *13*, 1692. [CrossRef]
16. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
17. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
18. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
19. Santosh, K.; Nattee, C. Template-based Nepali natural handwritten alphanumeric character recognition. *Sci. Technol. Asia* **2007** *12*, 20–30.
20. Moetesum, M.; Diaz, M.; Masroor, U.; Siddiqi, I.; Vessio, G. A survey of visual and procedural handwriting analysis for neuropsychological assessment. *Neural Comput. Appl.* **2022**, *34*, 9561–9578. [CrossRef]
21. Das, N.; Reddy, J.M.; Sarkar, R.; Basu, S.; Kundu, M.; Nasipuri, M.; Basu, D.K. A statistical–topological feature combination for recognition of handwritten numerals. *Appl. Soft Comput.* **2012**, *12*, 2486–2495. [CrossRef]
22. Sharma, A.K.; Thakkar, P.; Adhyaru, D.M.; Zaveri, T.H. Handwritten Gujarati character recognition using structural decomposition technique. *Pattern Recognit. Image Anal.* **2019**, *29*, 325–338. [CrossRef]
23. Mukherji, P.; Rege, P.P. Shape feature and fuzzy logic based offline devnagari handwritten optical character recognition. *J. Pattern Recognit. Res.* **2009**, *4*, 52–68. [CrossRef] [PubMed]
24. Itseez. Open Source Computer Vision Library. 2015. Available online: https://github.com/itseez/opencv (accessed on 15 January 2024).
25. Abadi, M.; Agarwal, A.; Barham, P. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. Available online: http://tensorflow.org/ (accessed on 7 March 2024).
26. Chollet, F., Keras. 2015. Available online: https://keras.io (accessed on 15 February 2024).
27. Bisong, E., Google Colaboratory. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*; Apress: Berkeley, CA, USA, 2019; pp. 59–64. [CrossRef]

28. LaValle, S.M.; Branicky, M.S.; Lindemann, S.R. On the relationship between classical grid search and probabilistic roadmaps. *Int. J. Robot. Res.* **2004**, *23*, 673–692. [CrossRef]
29. Iqbal, U.; Barthelemy, J.; Li, W.; Perez, P. Automating visual blockage classification of culverts with deep learning. *Appl. Sci.* **2021**, *11*, 7561. [CrossRef]
30. Iqbal, U.; Barthelemy, J.; Perez, P.; Davies, T. Edge-computing video analytics solution for automated plastic-bag contamination detection: A case from remondis. *Sensors* **2022**, *22*, 7821. [CrossRef]
31. Barthélemy, J.; Verstaevel, N.; Forehead, H.; Perez, P. Edge-computing video analytics for real-time traffic monitoring in a smart city. *Sensors* **2019**, *19*, 2048. [CrossRef] [PubMed]
32. Riaz, M.Z.B.; Iqbal, U.; Yang, S.Q.; Sivakumar, M.; Enever, K.; Khalil, U.; Ji, R.; Miguntanna, N.S. SedimentNet—A 1D-CNN machine learning model for prediction of hydrodynamic forces in rapidly varied flows. *Neural Comput. Appl.* **2023**, *35*, 9145–9166. [CrossRef]
33. Iqbal, U.; Barthelemy, J.; Perez, P. Prediction of hydraulic blockage at culverts from a single image using deep learning. *Neural Comput. Appl.* **2022**, *34*, 21101–21117. [CrossRef]