KurdSet Handwritten Digits Recognition Based on Different Convolutional Neural Networks Models

Sardar Hasen Ali [1], Maiwan Bahjat Abdulrazzaq [1]

[1] *Computer Science Department, University of Zakho, Duhok, Kurdistan Region, Iraq*

Abstract – Recognition of handwritten digits has garnered significant interest among researchers in the domain of recognizing pattern. This interest stems from the recognition's relevance in various real-life applications, including reading financial checks and official documents, which has remained a persistent obstacle. To address this challenge, researchers have developed numerous algorithms focusing on recognizing handwritten digits across different human languages. This paper presents a new Kurdish Handwritten dataset, consisting of Kurdish characters, digits, texts, and symbols. The dataset consists of 1560 participants, encompassing a broad and varied group. It serves as the primary dataset for training and evaluating algorithms in Kurdish digit recognition. We used Kurdish dataset named (KurdSet) and Arabic dataset for handwritten recognition, which holds 70,000 images of Arabic digits that were written by 700 various participants. Additionally, various models are utilized in the study, including ResNet50, DenseNet121, MobileNet, and a custom CNN (convolutional neural network). Additionally, the models' effectiveness was assessed through the examination of test accuracy, which measures the percentage of correctly classified digits in the evaluation phase.

ResNet50 also performs exceptionally well that achieved test accuracy 99.67%, indicating its All models exhibit good performance, DenseNet121 and the Custom CNN Model demonstrate the highest test accuracy of 99.73%, highlighting their superior performance. capabilities in capturing relevant features. Despite its accuracy, MobileNet still exhibits good recognition capability with a test accuracy 99.54%.

Keywords **– Deep neural network, custom CNN, DenseNet121, ResNet50, MobileNet, Kurdish handwritten digits dataset, Arabic handwritten digits dataset.**

DOI: 10.18421/TEM131-23
https://doi.org/10.18421/TEM131-23

Corresponding author: Sardar Hasen Ali
Computer Science Department, University of Zakho, Duhok, Kurdistan Region, Iraq
Email: sardar.ali@uoz.edu.krd

Received: 25 August 2023.
Revised: 13 November 2023.
Accepted: 12 December 2023.
Published: 27 February 2024.

1. Introduction

Nowadays, the recognition of handwritten digits is a dynamic area of exploration within the field of handwriting recognition. Numerous systems for recognizing handwritten digits have been put forth recently to meet the requirements of practical applications, emphasizing the need for high accuracy and reliability in recognition [1]. Handwriting digit recognition is the method that utilizes a computer or machine learning model to recognize and interpret handwritten digits, typically in the range of 0 to 9. This represents a crucial domain of investigation within the domains of machine learning and artificial intelligence, as it has numerous applications, including in postal automation, check processing, and signature verification. The process typically involves capturing an image of the handwritten digit, which is then preprocessed to remove noise and enhance contrast [2]. The preprocessed image is then linked up with a machine learning algorithm, having undergone training on an extensive dataset of manually written numerals, to predict the digit that has been written.

Various methods, including artificial neural networks, support vector machines, and k-nearest neighbor algorithms are utilized for the recognition of handwritten digits.

Notably, deep learning techniques, particularly convolutional neural networks, have demonstrated exceptional effectiveness and have achieved top-tier performance on widely used benchmark datasets [3]. Despite the challenges posed by handwriting variability, noise, and distortion in images, significant progress has been made in recent years in the field of handwritten digit recognition. This technology is now mature and highly accurate [3]. In another study [4], researchers developed a system for recognizing digit numerals in Arabic, English, Devanagari, and Persian languages. The system extracted structural characteristics of the numerals utilizing neural network and naïve Bayes classifiers on geometric primitives within images, resulting in high levels of accuracy.

Convolutional neural networks (CNNs) are pivotal in visual data analysis, excelling in image-related tasks [5], [6]. CNNs consist of interconnected layers for specialized operations. Feature maps capture patterns. Pooling layers reduce dimensions while retaining vital data using max pooling [7]. Fully connected layers aggregate features for predictions. They're trained via backpropagation with labeled data, proficient at automatic feature extraction, especially in computer vision [8]. On the other hand, DenseNet121, part of DenseNet model family introduced in 2017, excels in image classification [9]. Its design involves feed-forward connections between layers. Convolutional and dense blocks characterize it, efficiently reusing features. Transition layers manage complexity by reducing dimensions, performing well on benchmarks like ImageNet [10]. ResNet50, a ResNet variant with 50 layers, is renowned for image recognition [11]. Identity the proposed methods and the last section explains the summary of the work and then conclusion.

2. Theoretical Background

A hypothetical summary of the approaches, as proposed previously, is provided in the following subsections, aiming to acquaint readers with these techniques.

2.1. Convolutional Neural Networks

CNN is considered as one of the algorithms used in deep learning. Its primary advancement over previous approaches lies in its capacity to automatically detect relevant features without human intervention, drawing inspiration from the visual system of the human brain. This design enables CNNs to facilitate computer perception aligned with human cognition, thereby supporting tasks such as Processing of natural language, categorization of images, and identification of images [15].

shortcuts enable deep feature learning, aiding in complex pattern representation. It won ILSVRC 2015 [12]. MobileNet optimizes CNNs for mobile/embedded devices via depthwise separable convolutions [13]. It splits convolutions into depthwise and pointwise steps, enhancing efficiency [14].

This paper primarily presents the key contributions as follows:

Arabic and Kurdish digits datasets lie in their shared digits despite having different participants. Arabic digits dataset is sourced from native Arabic speakers or written materials, while Kurdish digits dataset originates from a distinct group of Kurdish participants.

We created Kurdish dataset (KurdSet) as there is no Kurdish dataset available in existing literature. To test the performance of deep learning models in Kurdish dataset, as far there has been no study that has compared or evaluated the two methods, we utilized multiple CNN models such as Densenet121, ResNet50, MobileNet, and custom CNsN models.

The article aims to compare and contrast the performance, advantages, as well as to evaluate the significance of the proposed Kurdish digits dataset for digit recognition in Kurdish language contexts.

The other sections of the document are structured as following: section two investigates into the framework theory, while the third section outlines the current state of similar works in the field. Section four reports the methodology of the study while section five discusses the results and follows by discussion of

As a subtype of deep neural networks (DNNs), the CNN typically encompasses key layers including convolutional, activation (employing nonlinear functions), with different types of layers [16], [17]. The following are the CNN layers:

2.1.1. Convolutional Layer

The vital element of a CNN is the convolutional layer, responsible for executing the "convolution" operation that lends the CNN its identification. Furthermore, convolutional kernel serves as a fundamental aspect of numerous technologies of computer vision, exerting a substantial influence on CNN's functionality. This operation involves a set of convolutional filters, termed kernels, that are applied to the image. This process entails the utilization of a small matrix, the kernel or filter, which is convolved with the image using its values [8]. Every filter aligns to a unique matrix, executing the convolution process on the input image. The outcome of this convolution is the generation of feature maps output.

This process is encapsulated in the computation of feature map values, as exemplified in below equation.

$$F[r, c] = (I * K)[r, c] = \sum i \sum j K[i, j]I[r - i, c - j]$$

In this context, "I" stands for the input image, and "K" denotes the kernel. The outcome matrix is characterized by row and column indices, denoted as r and c, respectively.

2.1.2. Activation Layer

The subsequent layer to a convolutional layer is the activation layer, also referred to as the nonlinear layer. This layer's fundamental purpose is to introduce nonlinearity into a predominantly linear computational system. A concrete example of this concept is the utilization of the rectified linear unit (ReLU) activation layer in conjunction with a convolutional layer employed alongside a convolutional layer, which enhances the nonlinearity within input data. This is achieved by transforming all input values into positive numbers, as ReLU yields an output of 0 for negative inputs. Furthermore, ReLU offers a primary advantage over alternative activation functions in terms of computational efficiency, resulting in decreased computational overhead [7]. Consequently, utilizing ReLU within CNNs leads to improved speed and simplicity within the prevailing context. The mathematical expression for ReLU can be depicted in equation.

$$f(k) = \max(0, k)$$

In this context, the implication of the function is that the output, represented by f(k), is zero when k is negative, while it remains consistently constant for positive values.

2.1.3. Pooling Layer

In CNN architectures, convolution layers are commonly interspersed with subsampling or pooling layers in neural networks. Each feature map produced by the convolutional layer undergoes independent processing. Pooling operations aimed at preventing overfitting and constraining the number of extracted features, are applied to decrease the spatial dimensions of the feature map. Two prevalent pooling techniques include the maximum pooling method and the average pooling method [18]. Figure 1 illustrates the pooling layer with 2x2 filter:

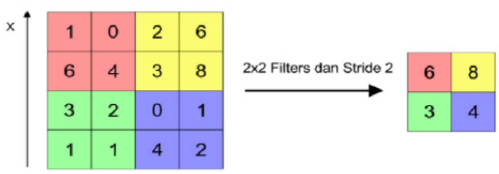

Figure 1. Pooling layer with 2x2 filter

2.1.4. Fully Connected Layer

While the convolutional and pooling layers offer insight into advanced attributes of source images, leveraging fully connected (FC) layers can notably become useful. Incorporating an FC layer presents a cost-effective means of acquiring arrangements of these characteristics that are not in a linear format. This FC layer generates a prospective classification score by amalgamating information from convolution and pooling layers, facilitating labeling of input images from the input layer. Subsequently, the two-dimensional outcome from the FC layer is directed to the output layer, where prediction of input class labels can be achieved through the application of a sigmoid function or softmax activation [19].

3. Related Work

CNNs have been highly successful for recognizing handwritten digits, often using the MNIST dataset as a benchmark. Earlier experiments frequently achieved accuracy rates of 100% on MNIST. In a study by Saptadeepa K. *et al*. [20], a direct approach for offline handwritten character recognition was employed. This encompassed the analysis of character geometry and the extraction of features based on gradients, using the EMNIST dataset. The dataset consisted of 240,000 images for training and 40,000 images for testing. The model demonstrated satisfactory digit recognition accuracy, which improved further through feature selection methods. Three techniques—mRMR, JMI, and Relief—selected pertinent features from the extracted feature vector. Results indicated the model's potential as an optimized solution for character recognition. Another study by Yaagoup *et al*. [21] focused on Arabic digit recognition, stressing its importance. Their model showcased exceptional performance on the MADBase dataset. Experiments yielded outstanding outcomes, including a remarkable 99.25% classification accuracy on the testing MADBase dataset. However, considerations regarding generalizability and scalability were noted.

Adhikary *et al.* [22] proposed a study for recognizing handwritten Dzongkha digits using machine learning techniques. Since no existing dataset was available, they manually created one using Google Jamboard and a digital pen tablet. This study is the first of its kind for the Dzongkha language. They utilized various algorithms, including SVMs, in addition to KNNs, and DTs as machine learning approaches for recognizing digits. SVM algorithm attained the utmost 98.29% accuracy, whereas the KNN algorithm reached 96.00% accuracy and the DT algorithm at 78.86% accuracy.

The authors acknowledged the recognition of Dzongkha digits poses challenges due to restricted resources and increased complexity when contrasted with recognizing English digit. Huda *et al.* [23] conducted a study where they trained a convolutional neural network model using two datasets named Bangla Lekha Isolated and NumbtaDB. The datasets were employed to identify digits and characters in the Bangla script. They implemented a shifting strategy to increase their dataset size and performed various experiments on vowels, numbers, and characters. The CNN model exhibited notable accuracies, achieving 96.42% on the Bangla Lekha dataset and 98.92% on the NumtaDB dataset. Furthermore, the researchers developed two practical applications based on their findings: a license plate recognition system and an intelligent e-learning platform that utilizes the trained model for recognizing Bangla script. Even though high accuracies were achieved on the datasets used in the studies, the models may not perform as well on other datasets or in real-world applications.

In a distinct study, a new technique for recognizing Arabic handwritten digits using deep learning methods was introduced [24]. The study aimed to tackle the challenge of Arabic digit recognition by employing the RBM-CNN (Restricted Boltzmann Machine - Convolutional Neural Network) approach. The author combined RBM for feature extraction with CNN for classification. Experimental outcomes demonstrated that this method surpassed existing approaches in terms of accuracy for Arabic digit recognition. The approach substantially elevated accuracy to 98.59%, outperforming previous methods and accomplishing the highest accuracy reported for the dataset. It is important to note that the study solely evaluated this approach on a single dataset, lacked comparison to other state of art techniques, and its applicability to real-world data remained uncertain.

Y. S. Can *et al.* [25] conducted a study to create an automated Arabic numeral spotting system for historical Ottoman population registers. They utilized a CNN-based approach with a red color mask, achieving 96.06% accuracy in numeral spotting. For digit recognition, their locally trained CNN outperformed DTL methods from modern datasets, which struggled with degradation in historical documents. The AHDBase dataset yielded better results than HODA dataset (Farsi handwritten digit dataset) due to digit similarity. However, CNN on the local dataset alone achieved 80% accuracy. In a separate study by Savita *et al.* [26], a hybrid model combining CNN for extraction and SVM for prediction was developed to recognize handwritten digits. Automatically generated CNN features proved superior to manually drawn ones. The hybrid model attained a high recognition accuracy of 99.28%. Though effective, the model's reliance on CNN and SVM, limited dataset testing, and uncertainty regarding real-world data performance call for further research. Comparisons with other methods and diverse settings are essential.

H. A. Jawad [27] introduced a deep learning approach for offline recognition of handwritten Arabic digits, employing CNN architecture to directly recognize digits from raw images without segmentation. The method achieved a validation accuracy of 94.3% on the AHDBase dataset, highlighting potential for improvement in hyperparameter tuning to address errors stemming from handwriting variations. Similarly, A. Pardamean *et al.* [28] assessed a CNN-based recognition system, employing the same architecture for training and testing on a dataset from AHDBase. Training involved 60,000 images over 250 epochs using backpropagation. Experimental outcomes revealed an average recognition rate of 97.67%, showing the model's strong performance in digit recognition.

R. H. Finjan [29] in his article underscores the importance of recognizing Arabic numerals and recommends the utilization of (CNNs) employing the ResNet-34 architecture. The research involved a dataset comprising 60,000 Arabic handwritten digits, with 1000 testing samples converted to grayscale to facilitate training. The dataset underwent preprocessing and augmentation, incorporating techniques for example, scaling, rotation, and shifting for enhancing diversity and size. The study's findings reveal that the proposed model achieves accurate recognition of Arabic handwritten digits, boasting a classification accuracy of 99.6%. These results affirm the efficiency of employing CNNs along with the ResNet-34 structure in the recognition of digits in Arabic handwritten, thereby making a valuable contribution to the domain of handwriting Arabic recognition.

AKM Ashiquzzaman and Abdul Kawsar Tushar developed a model utilizing CNN to identify handwritten Arabic digits. They evaluated their model and reached an accuracy of about 97.4% [30]. H. A. Alwzwazy [31] undertook a study involving an Arabic digit dataset collected from various school levels, meticulously prepared for analysis. Using this dataset, a CNN was trained, a potent deep learning model for image recognition. Rigorous parameter tuning and performance analysis ensured the CNN's robustness for the challenging dataset. Experiments utilized the caffe deep learning framework and high-performance GPUs, facilitating real-time applications. Impressively, the CNN achieved an outstanding 95.7% accuracy on the test dataset, confirming its efficacy for Arabic digit recognition.

While these studies demonstrated high accuracy on specific datasets, the generalization to diverse datasets and real-world handwriting variations remains unclear. Developing models that can adapt to such variations is crucial for broader applicability.

4. Methodology

This section outlines the methodology employed for crafting the suggested model. Section 4.1 provides an outline of Kurdish dataset, followed by the subsequent data preprocessing in the next step. Section 4.3 clarifies the design of the proposed CNN.

4.1. Dataset

We have developed a newly generated dataset known as Kurdish Handwritten dataset called (KurdSet), which comprises three pages of content consisting of characters, digits, texts, and symbols in the Kurdish language. Kurdish is the language spoken in the Kurdistan region. The dataset was compiled with the involvement of both males and females, ranging in age from 18 to 65 years. The participants encompassed individuals from various professional backgrounds, including employees, and students from diverse departments. They were affiliated with the University of Zakho, all departments including school of languages and scientific departments, University of Duhok, Technical Institute of Zakho, and Duhok. Approximately 2000 users took part in the dataset creation process; however, certain forms were written outside the designated boxes or were unclear, so we excluded or disregarded those entries, resulting in a final count of 1560 participants. Figure 2 shows the Kurdish digit dataset sample.

For this article we choose the digits from Kurdish handwritten dataset which ranged from (0 to 9) in order to apply to Kurdish handwritten digits and Arabic handwritten digit recognition database. Kurdish dataset includes 15,600 images of Kurdish number. These images are often labeled with the corresponding digit for training and evaluation purposes in machine learning and computer vision tasks. These images are manually annotated with corresponding labels indicating the digit they represent. The database is commonly utilized for the purpose of training and evaluation algorithms and models are used for recognizing Kurdish digits in handwritten documents. On the other hand, the recognition of Arabic handwritten digits dataset contains 70,000 images written by 700 participants. Each participant has provided ten samples for every Arabic digit (0 to 9), leading to a range of variations and diverse writing styles for each participant.

Arabic handwritten digit information for the database was gathered from diverse establishments, encompassing various schools and university sections such as, medicine, engineering, and other different institutions. Language writers present the main difference between these two datasets for Kurdish handwritten dataset participants are Kurdish participants whose native language is Kurdish while for Arabic handwritten digit dataset participants are Arabs. Figure 3 shows Kurdish dataset (KurdSet) as well as the architecture design of Kurdish handwritten dataset in a form of a flow chart.

4.2. Data Preprocessing

As for preprocessing techniques, we applied various methods, techniques, and operations to input images of handwritten digits before they are analyzed by a recognition system. Its purpose is to improve image quality and extract important features for accurate recognition. In the case of our dataset, Kurdish handwritten dataset (KurdSet), we implemented specific preprocessing steps. We resized the image size to 80×80 and converted the background to white to reduce noise. Additionally, we converted the images to grayscale, performed cropping and centering to ensure the important content is properly aligned and focused. The models that are utilized in this paper are ResNet50, DenseNet121, MobileNet and Custom CNN model. The metrics used in ResNet50, DenseNet121, and MobileNet models are accuracy, F1, AUC, we used freezing specific layers' weights and pre-trained models in transfer learning on imageNet with learned features from large datasets. By preserving lower layers responsible for general low-level features, the focus can shift to fine-tuning higher layers specialized for the task at hand. This strategy conserves computational resources and mitigates overfitting, particularly when training data is scarce. Additionally, we added two layers to these models and trained the weights of these later layers, in order to capture the high-level features learned by the model before the final classification layer. This can be useful for tasks such as feature extraction, transfer learning, or any other scenario where we want to utilize the intermediate representations of the model. There are 10 classes as an output and hyper parameter that is used along with these models. The models are compiled with the following settings: loss function employed is categorical crossentropy, and the optimizer utilized is Adam, and accuracy is used as the metric. The training process consists of 50 epochs. For the last layer we utilized softmax activation function to classify the classes.

4.3. The Proposed of CNN

The Kurdish handwritten dataset is divided into three primary sets: 80% is utilized for training phase, 10% for testing phase, and other 10% for validation phase. The custom CNN model being suggested comprises six convolutional layers and one dense layer, and utilizes ReLU activation and max pooling. It has a final softmax activation for classifying the 10 output classes.

The model employs a loss function of cross-entropy category and utilizes Adam optimizer and accuracy metric. F1 score assesses overall accuracy considering precision and recall, while AUC measures discriminative power. This custom CNN model is applied to Arabic and Kurdish handwritten digit recognition, with a depicted flowchart in Figure 4 showcasing the method.

Figure 2. Kurdish digit dataset sample

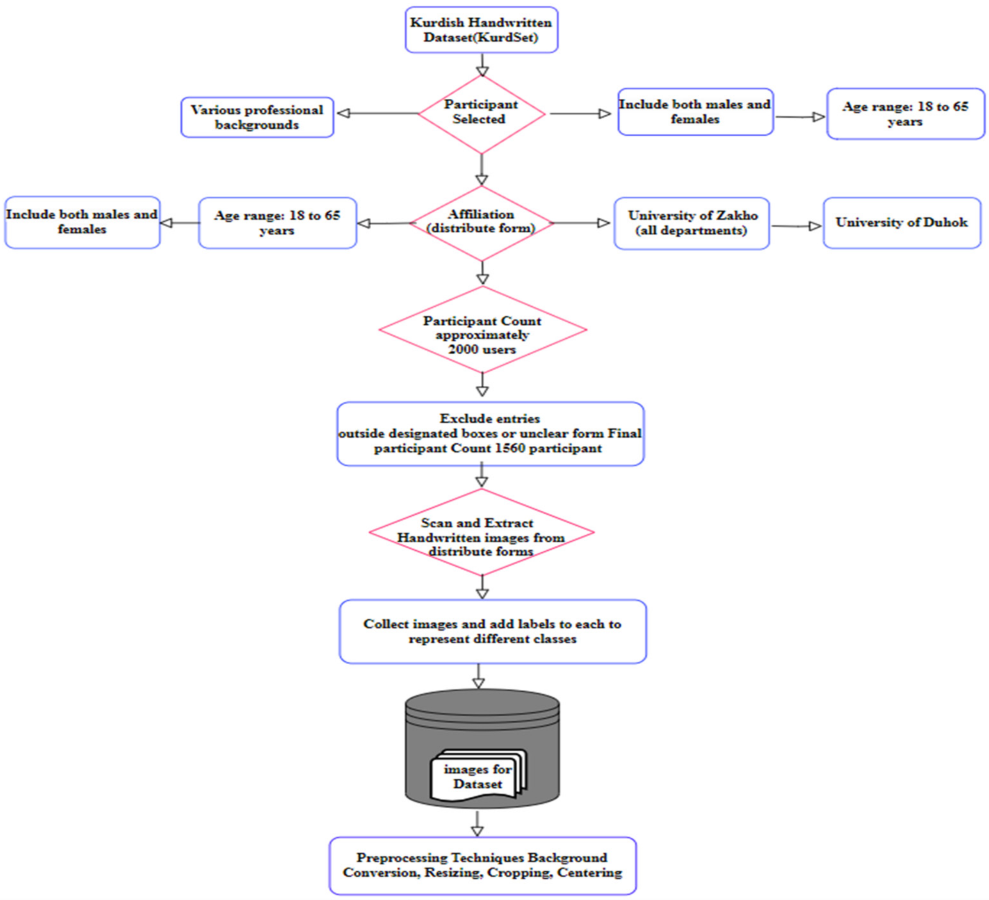

Figure 3. The architectural design of Kurdish (KurdSet) handwritten dataset in the form of a flowchart

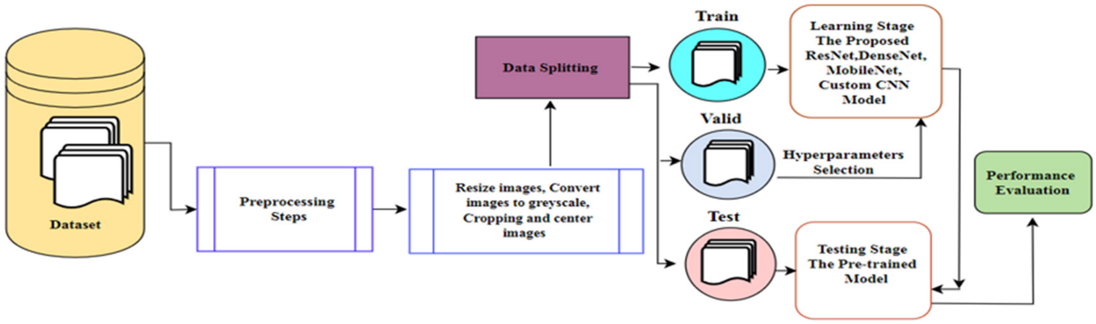

Figure 4. The flowchart of the method

5. Results and Discussion

In this section, we conducted a thorough evaluation. The models were implemented using Python programming language. We utilized two datasets for evaluation purposes: the Arabic handwritten digits dataset and our newly created dataset called Kurdish handwritten dataset. Four models, namely Densenet121, ResNet50, MobileNet, and Custom CNN, are applied to these datasets. For both datasets we accomplished these equations for accuracy, training loss and validation loss. In classification tasks, accuracy is calculated as follows:

$$\text{Accuracy}=\frac{\text{Number of correctly predicted instances}}{\text{Total number of instances}}\times 100\%$$

The formula used for calculating the training loss is:

Training Loss=(1/N)*∑(Loss(predicted, ground_truth))

N represents the total number of samples in the training dataset.

Loss (predicted, ground_truth) is the loss function that computes the error in the predicted values and the true target labels for each sample.

While the formula for validation loss is: validation loss = (1 / N) * ∑(loss(predicted, ground_truth))

Where: N forms the total number of samples in validation dataset. Loss (predicted, ground_truth) that is previously referred to. Results and discussions of each model are presented in the following subsections.

5.1. Kurdish Handwritten Dataset (KurdSet)

In the training phase of a classifier algorithm, accuracy and loss are commonly used metrics that are tracked over successive epochs to prevent overfitting and evaluate the performance of predictions. These metrics are believed to have an inverse relationship, meaning that low loss values are typically associated with high accuracy values.

The Kurdish handwritten dataset is ideal for training and evaluating handwritten digit recognition due to its diversity aiding effective learning. Employing the DenseNet121 model led to remarkable outcomes. The train accuracy reached 99.95% and while valid accuracy reached 99.70% highlighting its proficiency in accurate classification. This underscores its adeptness in learning from the dataset, successfully recognizing Kurdish script digits.

Train loss is 0.0014 and valid loss is 0.0183, demonstrating error minimization during training and consistent performance. A 99.73% test accuracy affirms strong learning and feature capture. Figures 5 and 6 further illustrate model performance.

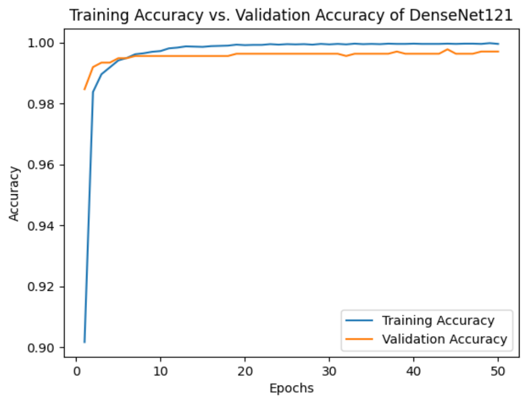

Figure 5. The training accuracy in comparison to validation accuracy for the DenseNet121

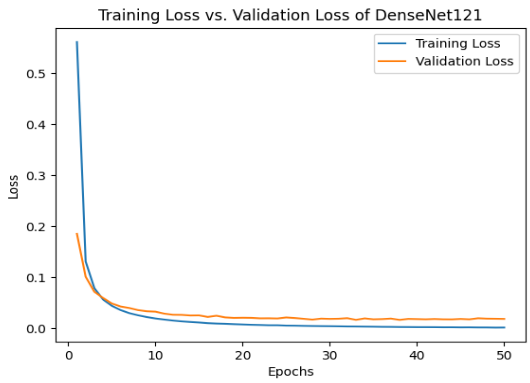

Figure 6. The progression of training loss compared to validation loss in the DenseNet-121 model

ResNet50 model excels academically in recognizing handwritten digits, achieving 99.99% training and 99.56% validation accuracy. Its architecture, using residual connections, captures intricate patterns for image recognition. With low training loss (0.0004) and acceptable validation loss, it demonstrates effective learning and generalization on the Kurdish handwritten dataset. Test accuracy of 99.67% further validates its excellence. Figures 7 and 8 provide visual proof.

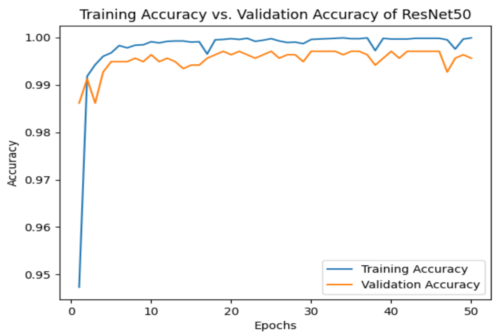

Figure 7. The transition of training accuracy versus validation accuracy of ResNet50 model

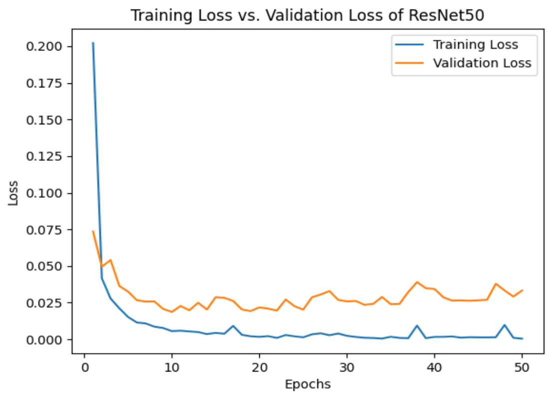

Figure 8. The transition of training versus validation loss of ResNet50 model

MobileNet model performs exceptionally well on the Kurdish handwritten dataset, achieving a remarkable accuracy for the test that reached 99.54%. Its accuracy for training was 99.98% and validation accuracy was 99.70% which underscore its adeptness at learning and recognizing patterns in Kurdish handwritten digits. These scores demonstrate its proficiency in digit recognition. Known for efficiency, MobileNet is ideal for mobile and embedded applications, balancing size and accuracy effectively. With a training loss of 0.0004 on the Kurdish handwritten dataset, the model effectively learns patterns, leading to a high-test accuracy of 99.54%. Figures 9 and 10 show the results.

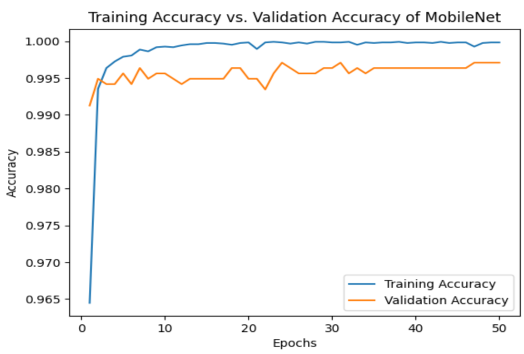

Figure 9. The transition of training versus validation accuracy of MobileNet

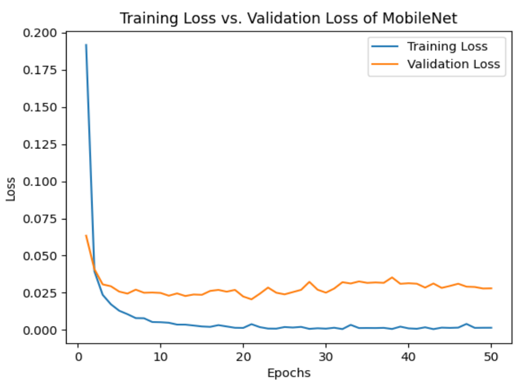

Figure 10. The transition of training versus validation Loss of MobileNet

Custom CNN model achieves impressive results on Kurdish Handwritten dataset for digit recognition. Training accuracy of 99.56% demonstrates effective learning of dataset patterns, and 99.49% validation accuracy showcases consistent performance on new data. The training loss of 0.0474 signifies successful pattern learning, while a validation loss of 0.2019 represents model performance. Despite slightly higher validation loss, reasonable generalization is evident. Overall, the custom CNN model effectively learned dataset patterns and features. With high performance, it attains a test accuracy of 99.73%, comparable to the DenseNet121 model. Graphs 11 and 12 illustrate these results.

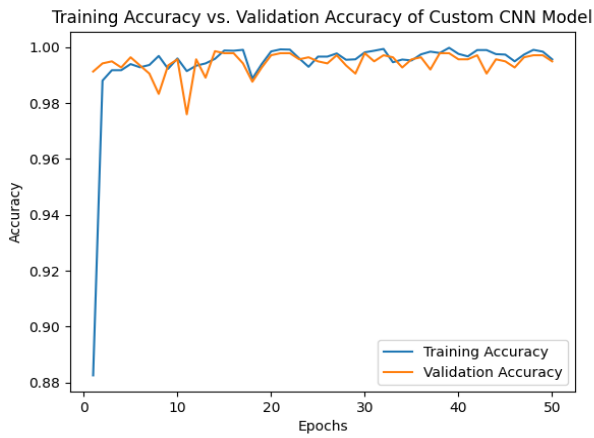

Figure 11. The transition of training versus validation accuracy of Custom CNN Model

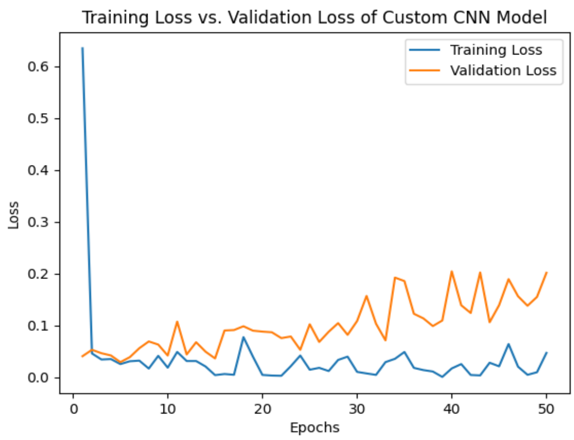

Figure 12. The transition of training versus validation Loss of Custom CNN Model

DenseNet121 and custom CNN model have the highest test accuracy, showing superior performance. ResNet50 performs well, highlighting its feature-capturing capabilities. Despite slightly lower accuracy, MobileNet still exhibits good recognition. These models demonstrate unique strengths in capturing patterns, resulting in varying test accuracy on the Kurdish handwritten dataset. See Table 1 for detailed performance.

Table 1. The models results of Kurdish handwritten dataset

Models	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss	Test Accuracy
DenseNet121	99.95%	99.70%	0.0014	0.0183	99.73%
ResNet50	99.99 %	99.56%	0.0009	0.0332	99.67%
MobileNet	99.98%	99.70%	0.0014	0.0279	99.54%
Custom CNN Model	99.56%	99.49%	0.0474	0.2019	99.73%

The F1 score and AUC values of all models on the KurdSet dataset indicate outstanding performance in recognizing Kurdish handwritten digits. DenseNet121 achieves high scores, while ResNet50 performs even better with a perfect AUC of 1.0. MobileNet also demonstrates proficiency in classification, and the Custom CNN Model achieves perfect scores across all metrics. Overall, these models exhibit exceptional achievement and reliability in accurately classifying Kurdish handwritten digits, showing their effectiveness in distinguishing between different digit classes in the dataset. Table 2 provides the outcomes of F1 and AUYC.

Table 2. The outcomes of F1 and AUS models applied Kurdish handwritten dataset

Models	F1	AUC
DenseNet121	0.996769	0.999996
ResNet50	0.998022	1.0
MobileNet	0.996071	0.999992
Custom CNN Model	1.0	1.0

5.2. *Arabic Handwritten Digits Dataset*

Arabic handwritten digit recognition using AHDBase database yields high accuracy. DenseNet121 achieves 99.98% training accuracy, precisely learning dataset patterns, and 99.53% validation accuracy, ensuring consistency with new data. DenseNet121's architecture stands out, delivering strong results. Low training loss (0.0007) minimizes errors, while validation loss (0.021) maintains consistency with unseen data. This well-trained model, despite a slightly higher validation loss, demonstrates good generalization, characteristics contributing, an exceptional 99.64% test accuracy for Arabic handwritten digits is achieved, as depicted in Figures 13 and 14.

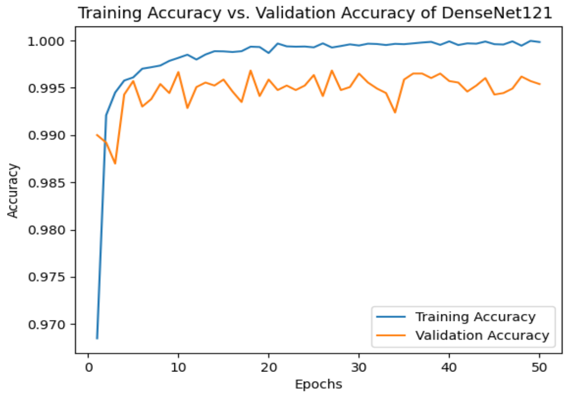

Figure 13. The transition of training accuracy versus validation accuracy of DernseNet121

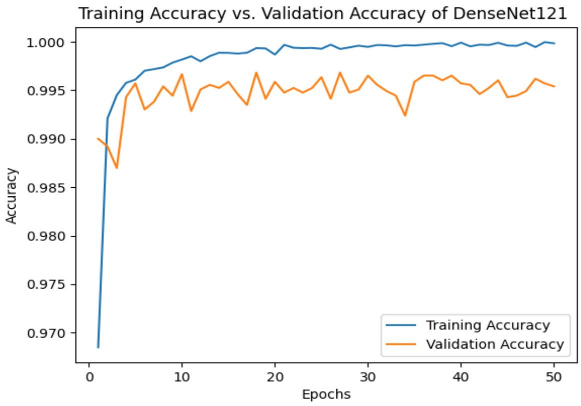

Figure 14. The transition of training loss versus validation Loss of DernseNet12

Figure 15 shows ResNet50's Arabic handwritten digit recognition results, with 1.000% training accuracy and 99.73% validation accuracy. The model successfully learns features demonstrating effective recognition. ResNet50's depth and skip connections capture complex patterns for high accuracy. Figure 16 clarifies findings, showing training loss (3.0369) and validation loss (0.0197), while ResNet50 achieves an outstanding 99.74% test accuracy on the Arabic dataset.

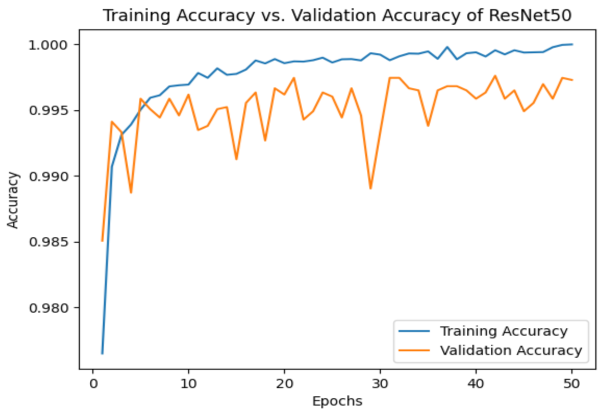

Figure 15. The transition of training accuracy versus validation accuracy of ResNet50

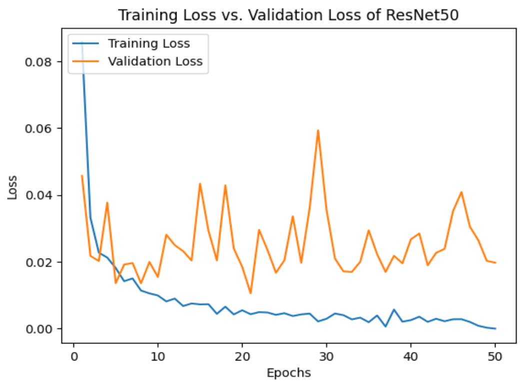

Figure 16. The transition of training loss versus validation Loss of ResNet50 Model

The MobileNet model results, depicted in Figures 17 and 18, are high. It achieves a high training accuracy of 99.98%, demonstrating effective learning of Arabic handwritten digit patterns and features. With a validation accuracy of 99.44%, the model generalizes well to unseen data, recognizing various writing styles. Training loss of 2.4349 reflects training data fitting, while validation loss of 0.1365 shows unseen data performance. These values reveal the model's capacity to understand the data.

Overall, the MobileNet model's high training accuracy, strong validation accuracy, and low losses indicate its successful pattern and feature learning, leading to a remarkable test accuracy of 99.48%.

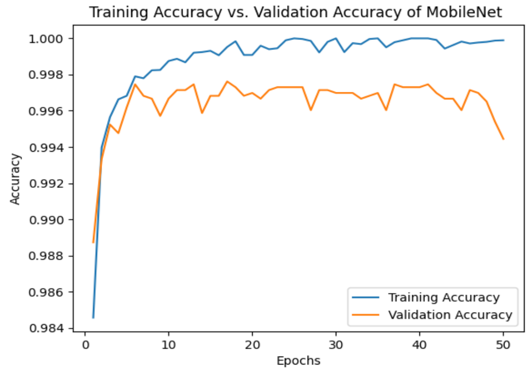

Figure 17 shows *the transition of training accuracy versus validation accuracy of MobileNet model*

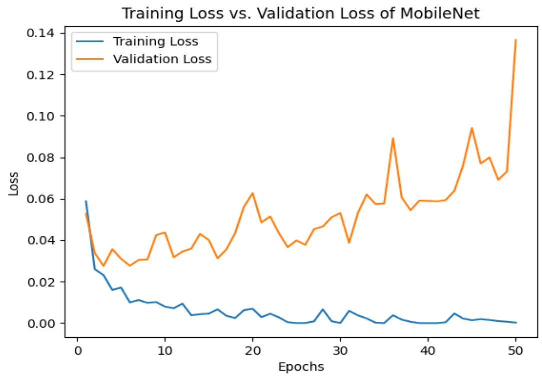

Figure 18. The transition of training loss versus validation Loss of MobileNet

CNN model that has six convolutional layers, 1 dense layer, ReLU activation, and max pooling, while The Custom CNN model, featuring 6 convolutional layers, 1 dense layer, max pooling, and ReLU activation, effectively extracts features from Arabic handwritten digits. It achieves high training accuracy (99.82%) and impressive validation accuracy (99.52%), showcasing strong learning and generalization.

Table 3. The models results of Arabic handwritten dataset

Models	**Training Accuracy**	**Validation Accuracy**	**Training Loss**	**Validation Loss**	**Test Accuracy**
DenseNet121	99.98%	99.53%	0.0007	0.0219	99.64%
ResNet50	1.0000%	99.73%	0.0002	0.0197	99.74%
MobileNet	99.98%	99.44%	0.0003	0.1365	99.48%
Custom CNN Model	99.82%	99.52%	0.0098	0.0702	99.65%

The model's architecture captures intricate features, resulting in a remarkable 99.65% test accuracy. Figures 19 and 20 show the results.

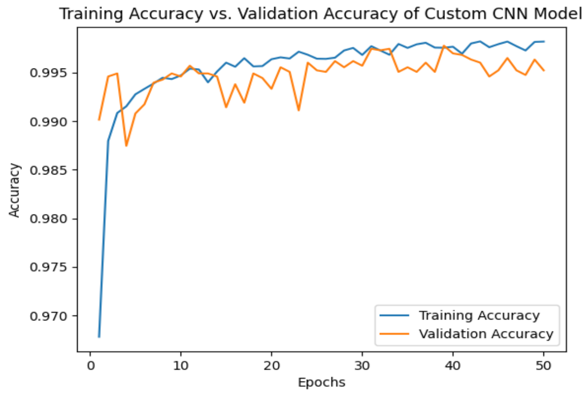

Figure 19. The transition of training accuracy versus validation accuracy of Custom CNN

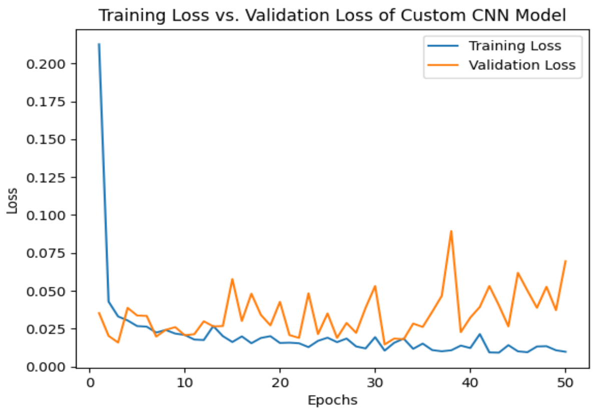

Figure 20. The transition of training loss versus validation Loss of Custom CNN

All models exhibit good performance, with ResNet50 achieving the highest test accuracy. This can be attributed to its deep residual connections and capability to handle complex networks, allowing it to outperform the other models. DenseNet121 and the custom CNN model also deliver strong performance, leveraging their respective architectures to effectively capture relevant features. MobileNet, although slightly lower in accuracy, remains efficient and proficient in recognizing Arabic handwritten digits. Overall, each model demonstrates unique strengths in capturing patterns and features, resulting in varying levels of test accuracy. Table 3 depicts Arabic handwritten dataset.

Additionally, the F1 score and AUC metrics provide comprehensive insights into the performance of classification models. The high F1 scores indicate that the models excel in correctly classifying positive and negative instances while the AUC metric evaluates their ability to distinguish between classes. The provided data demonstrates that all models achieve high F1 scores and AUC values, showcasing their effectiveness in accurately classifying Arabic handwritten digits. This indicates their strong performance and reliability in the Arabic handwritten digit dataset. The outcomes are provided in Table 4.

Table 4. The outcomes of F1 and AUS models applied on Arabic handwritten digits dataset

Models	**F1**	**AUC**
DenseNet121	0.9962854	0.9998708
ResNet50	0.9945733	0.9999133
MobileNet	0.9960017	0.9999526
Custom CNN Model	0.9951455	0.9999719

5.3. Comparison with Previous Works

When contrasting these findings with the latest and most advanced methodologies in [27], [30], the proposed models showed superior performance. For instance, the state of art in [27] and [30] achieved accuracies of 94.30% and 97.40%, respectively. Furthermore, in [24] the accuracy result is 98.59% using RBM-CNN method, whereas in [29], the ResNet-34 model attained an accuracy of 99.60%. Table 5 shows the outcomes of the proposed method with the state of art approaches.

Table 5. Assessing the accuracy of the proposed approach comparing with other state of art approaches

Models	**Dataset**	**Accuracy**
RBM-CNN[24]	AHDBase	98.59%
Hybrid CNN [27]	MNIST+AHDBase	94.30%
ResNet-34[29]	AHDBase	99.60%
CNN[30]	AHDBase	97.40%
Proposed Custom CNN	AHDBase	99.65%
Proposed ResNet50	AHDBase	99.74%
Proposed MobileNet	AHDBase	99.48%
Proposed DenseNet121	AHDBase	99.64%

6. Conclusion

The conclusion analysis presented in the article highlights the strong performance and test accuracy of different models in comparing Arabic and Kurdish handwritten digits datasets. ResNet50, with its deep residual connections and ability to handle complex networks, achieved the highest test accuracy on the Arabic dataset. DenseNet121 and the custom CNN model also delivered impressive results, effectively capturing relevant features. Although MobileNet had a slightly lower accuracy, it remained efficient in recognizing Arabic handwritten digits. These models outperformed state-of-the-art approaches, demonstrating their superiority in classification tasks. On the other hand, on the Kurdish handwritten digit dataset, DenseNet121 emerged as a top-performing model, achieving high training and validation accuracies. ResNet50, MobileNet, and the custom CNN model also showed strong performance in recognizing Kurdish handwritten digits. These models effectively learned the patterns and features of the dataset, enabling accurate predictions and demonstrating their potential in real-world applications.

Each model demonstrated its unique strengths, such as deep residual connections, efficiency, or effective feature extraction. The findings contribute to the domain of pattern recognition, providing valuable perspectives for improving classification algorithms and offering promising avenues for future research in handwritten digit recognition.

Acknowledgment

We express our gratitude to the University of Zakho for their support, which enabled us to conduct our research and finish it successfully, especially computer science department.

References:

[1]. Agrawal, A. K. (2021). Design of CNN based model for handwritten digit recognition using different optimizer techniques. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *12*(12), 3812-3819.

[2]. Shamim, S. M., Miah, M. B. A., Sarker, A., Rana, M., & Jobair, A. Al. (2018). Handwritten digit recognition using machine learning algorithms. *Indonesian Journal of Science and Technology, 3*(1), 29–39. Doi: 10.17509/ijost.v3i1.10795

[3]. Upender K. & Pasupuleti, V. S. K. (2021). Real Time Handwritten Digits Recognition Using Convolutional Neara Network. *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE).*

[4]. Al-Hmouz, A., Latif, G., Alghazo, J., & Al-Hmouz, R. (2020). Enhanced numeral recognition for handwritten multi-language numerals using fuzzy set-based decision mechanism. *International Journal of Machine Learning and Computing*, *10*(1), 99-107.

[5]. Sharma, M., Sindal, P.S., Baskar, M. (2023). Handwritten Digit Recognition Using Machine Learning. In Khanna, A., Polkowski, Z., Castillo, O. (eds) *Proceedings of Data Analytics and Management. Lecture Notes in Networks and Systems, 572*. Springer, Singapore. Doi: 10.1007/978-981-19-7615-5_3

[6]. Jain, H., Sharma, N. (2021). Handwritten Digit Recognition Using CNN. In Goyal, D., Gupta, A.K., Piuri, V., Ganzha, M., Paprzycki, M. (eds) *Proceedings of the Second International Conference on Information Management and Machine Intelligence. Lecture Notes in Networks and Systems, 166*. Springer, Singapore. Doi: 10.1007/978-981-15-9689-6_69

[7]. Baldominos, A., Saez, Y., & Isasi, P. (2019). A survey of handwritten character recognition with MNIST and EMNIST. *Applied Sciences (Switzerland), 9*(15). Doi: 10.3390/app9153169

[8]. Haghighi, F., & Omranpour, H. (2021). Stacking ensemble model of deep learning and its application to Persian/Arabic handwritten digits recognition. *Knowledge-Based Systems, 220*, 106940. Doi: 10.1016/j.knosys.2021.106940

[9]. Tapotosh Ghosh, Abedin, M. H. Z., Al Banna, H., Mumenin, N., & Abu Yousuf, M. (2021). Performance Analysis of State of the Art Convolutional Neural Network Architectures in Bangla Handwritten Character Recognition. *Pattern Recognition and Image Analysis, 31*(1),60–71. Doi: 10.1134/S1054661821010089

[10]. Mocsari, E., & Stone, S. S. (1978). Colostral IgA, IgG, and IgM-IgA fractions as fluorescent antibody for the detection of the coronavirus of transmissible gastroenteritis. *American Journal of Veterinary Research, 39*(9), 1442–1446.

[11]. Seng, L. M., Chiang, B. B. C., Salam, Z. A. A., Tan, G. Y., & Chai, H. T. (2021). MNIST handwritten digit recognition with different CNN architectures. *Journal of Applied Technology and Innovation, 5*(1), 7–10.

[12]. Sangeetha, V., & Prasad, K. J. R. (2006). Syntheses of novel derivatives of 2-acetylfuro[2,3-a]carbazoles, benzo[1,2-b]-1,4-thiazepino[2,3-a]carbazoles and 1-acetyloxycarbazole-2- carbaldehydes. *Indian Journal of Chemistry - Section B Organic and Medicinal Chemistry, 45*(8),1951–1954. Doi: 10.1002/chin.200650130

[13]. Sayeed, A., Shin, J., Hasan, M. A. M., Srizon, A. Y., & Hasan, M. M. (2021). BengaliNet: A low-cost novel convolutional neural network for Bengali handwritten characters recognition. *Applied Sciences, 11*(15). Doi: 10.3390/app11156845

[14]. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

[15]. Sakib, S., Ahmed, N., Kabir, A. J., & Ahmed, H. (2019). An Overview of Convolutional Neural Network : Its Architecture and Applications. *Preprints* 2018. Doi: 10.20944/preprints201811.0546.v4

[16]. Alzubaidi, L., Zhang, J., Humaidi, A. J., Dujaili, A. Al, Duan, Y., Shamma, O. Al, Santamaría, J., Fadhel, M. A., Amidie, M. Al, & Farhan, L. (2021). Review of deep learning : concepts , CNN architectures , challenges , applications , future directions. *Journal of Big Data, 8*. Springer International Publishing. Doi: 10.1186/s40537-021-00444-8

[17]. Arif, R. B., Siddique, M. A. B., Khan, M. M. R., & Oishe, M. R. (2018, September). Study and observation of the variations of accuracies for handwritten digits recognition with various hidden layers and epochs using convolutional neural network. In *2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT)*, 112-117. IEEE.

[18]. Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, 818-833. Springer International Publishing.

[19]. Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*.

[20]. Kalita, S., Gautam, D., Kumar Sahoo, A., & Kumar, R. (2019). A combined approach of feature selection and machine learning technique for handwritten character recognition. *International Journal of Advanced Studies of Scientific Research*, *4*(4).

[21]. Yaagoup, K. M. M., & Hamid, A. E. (2020). Online Handwritten Arabic Digits (Indian) Recognition using Deep learning. *Ijarcce, 9*(11), 47–55. Doi: 10.17148/ijarcce.2020.91109

[22]. Adhikary, P. K., Dadure, P., Saha, P., Tawmo, & Pakray, P. (2023). Dzongkha Handwritten Digit Recognition using Machine Learning Techniques. *Procedia Computer Science, 218*, 2350–2358. Doi: 10.1016/j.procs.2023.01.210

[23]. Huda, H., Fahad, M. A. I., Islam, M., & Das, A. K. (2022). Bangla handwritten character and digit recognition using deep convolutional neural network on augmented dataset and its applications. In *2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, 1-7. IEEE. Doi: 10.1109/IMCOM53663.2022.9721634

[24]. Alani, A. A. (2017). Arabic handwritten digit recognition based on restricted Boltzmann machine and convolutional neural networks. *Information (Switzerland), 8*(4). Doi: 10.3390/info8040142

[25]. Can, Y. S., & Kabadayi, M. E. (2020). Automatic CNN-based arabic numeral spotting and handwritten digit recognition by using deep transfer learning in ottoman population registers. *Applied Sciences (Switzerland), 10*(16). Doi: 10.3390/APP10165430

[26]. Ahlawat, S., & Choudhary, A. (2020). Hybrid CNN-SVM Classifier for Handwritten Digit Recognition. *Procedia Computer Science, 167*(2019), 2554–2560. Doi: 10.1016/j.procs.2020.03.309

[27]. Alkhateeb, J. H. (2020). Handwritten Arabic Digit Recognition Using Convolutional Neural Network. *International Journal of Communication Networks and Information Security, 12*(3), 411–416. Doi: 10.17762/ijcnis.v12i3.4807

[28]. Pardamean A., Yuliana D., Watmah S., Hikmawan S., & Sfenrianto. (2020). Arabic Handwritten Digit Recognition using Convolutional Neural Network. *IJRTE, 8*(6).DOI:10.35940/ijrte.F7745.038620

[29]. Finjan, R. H., Rasheed, A. S., Hashim, A. A., & Murtdha, M. (2021). Arabic handwritten digits recognition based on convolutional neural networks with resnet-34 model. *Indonesian Journal of Electrical Engineering and Computer Science, 21*(1), 174–178. Doi: 10.11591/ijeecs.v21.i1.pp174-178

[30]. Ashiquzzaman, A., & Tushar, A. K. (2017, February). Handwritten Arabic numeral recognition using deep learning neural networks. In *2017 IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, 1-4. IEEE. Doi: 10.1109/ICIVPR.2017.7890866

[31]. Alwzwazy, H. A., Albehadili, H. M., Alwan, Y. S., & Islam, N. E. (2016). Handwritten digit recognition using convolutional neural networks. *International Journal of Innovative Research in Computer and Communication Engineering*, *4*(2), 1101-1106.