Handwritten digit recognition using CNN

Comparative Study of Machine Learning Algorithms

1st Rawaa Ayman Hammad
Computer Systems Engineering Department
BIRZEIT UNIVERSITY
RammAllah, Palestine
1210927@student.birzeit.edu

2nd Lina Nedal Abufarha
Computer Systems Engineering Department
BIRZEIT UNIVERSITY
RammAllah, Palestine
1211968@student.birzeit.edu

3rd Khalid Omari
Computer Systems Engineering Department
BIRZEIT UNIVERSITY
RammAllah, Palestine
1192973@student.birzeit.edu

Abstract—This project focuses on building a Handwritten Digit Recognition System capable of recognizing numbers written in both Arabic and English. The system uses three machine learning models: Decision Tree, K-Nearest Neighbors (KNN), and Convolutional Neural Networks (CNN). A graphical user interface (GUI) was implemented to provide a user-friendly way to upload images and get predictions.

Index Terms—Handwritten Digit Recognition, Machine Learning, Convolutional Neural Networks, K-Nearest Neighbors, Decision Tree, Arabic Digits, English Digits, Confusion Matrix, Deep Learning, Image Classification

I. Introduction

The process involves data preprocessing to prepare the datasets for analysis, training models using different algorithms, and evaluating their performance based on standard metrics. The main goal is to determine the best-performing model by comparing the accuracy, precision, recall, and F1score of each algorithm. The models tested in this project include Decision Tree, K-Nearest Neighbors (KNN), and Convolutional Neural Networks (CNN). This comparative analysis ensures that the developed system is efficient and capable of accurate digit recognition across different languages.

II. Dataset Description We used two datasets

for this project:

A. Arabic Digits Dataset:

Was obtained from the Arabic Handwritten Digit Dataset (AHDD) from Kaggle This dataset is widely used for Arabic handwritten digit recognition tasks, consists of 88,000 images, divided as follows: [1].

1- Training set: 60,000 images
2- Test set: 28,000 images

Image Dimensions: Each image is (28x28) pixels, represented as a 784-dimensional vector (flattened) [2].

The dataset is stored in CSV format:

csvTrainImages 60k x 784.csv: Training images. csvTrainLabel 60k x 1.csv: Labels for training images. csvTestImages 10k x 784.csv: Testing images. csvTestLabel 10k x 1.csv: Labels for testing images.

B. English Digits Dataset (MNIST):

The dataset is the widely recognized MNIST Handwritten Digit database. It is preloaded in Python using TensorFlow/Keras.

1- Training Images: 60,000 samples
2- Testing Images: 10,000 samples

Image Dimensions: 28x28 pixels, Contains numeric labels from 0 to 9.

Preprocessing: Each image were normalized to have pixel values in the range [0, 1]. Labels were one-hot encoded for compatibility with neural network models.

III. Algorithms Used

A. Decision Tree:

The Decision Tree algorithm is a supervised learning method used for classification and regression tasks. In this project, it was implemented to classify handwritten digits based on pixel values from the dataset. The algorithm works by recursively splitting the dataset into branches based on feature thresholds. Each split is designed to maximize information gain or minimize impurity, such as Gini index or entropy. At each decision node, the algorithm evaluates specific pixel intensities to determine the branch leading to the correct classification. This method is simple yet effective for smaller or less complex datasets, although it can become prone to overfitting with large, high-dimensional data like handwritten digits.

B. K-Nearest Neighbors (KNN):

K-Nearest Neighbors (KNN) is a non-parametric, instancebased learning algorithm that classifies a given input by considering the majority class of its nearest neighbors in the feature space. For this project, the KNN algorithm was trained with k=3, meaning it considers the three closest neighbors to predict the class of a handwritten digit. Distance metrics, such as Euclidean distance, are used to identify these neighbors. KNN performs well with relatively small datasets and is effective when the data is evenly distributed across classes. However, it requires computational resources proportional to the size of the dataset during the prediction phase, as it involves calculating distances for each new input.

C. *Convolutional Neural Network (CNN):*

The Convolutional Neural Network (CNN) is a deep learning algorithm specifically designed to analyze visual data such as images. In this project, CNN was utilized to classify handwritten digits in both Arabic and English datasets. The CNN architecture consists of:

Convolutional Layers: These extract essential features such as edges, curves, and shapes from the input images.

Pooling Layers: These reduce the spatial dimensions of the feature maps, making the model more computationally efficient while retaining the most relevant features.

Fully Connected Layers: These layers interpret the extracted features and map them to the respective digit classes.

IV. Model Training and Evaluation

In Building the CNN Model We added Conv2D layers to extract features from the images using filters. MaxPooling2D was used to reduce the spatial dimensions of the data while retaining important features. Dropout layers were added to prevent overfitting. The Flatten layer converted the data into a 1D array to pass it to dense layers. Finally, Dense layers were added for the final processing, with Softmax activation for the output layer.

a) Model Configuration:: We used the Adam optimizer to optimize model performance. The loss function categorical crossentropy was chosen because it is suitable for multi-class classification. Accuracy was added as a metric to evaluate the model during training.

b) Training the Model:: The model was trained on the training data for 5 epochs with a batch size of 32. Validation data was used to evaluate the model after each epoch.

c) Saving the Model:: After training, the model was saved to a file named cnnmodel.h5, so it can be reused later for predictions

We trained the models on both Arabic and English datasets and evaluated their performance using the following metrics: Accuracy: measures the percentage of correct predictions out of the total predictions. Training Time: the time required to train each model.

A. *Accuracy Analysis:*

CNN outperformed other models in both Arabic and English datasets, achieving the highest accuracy due to its ability to learn spatial features. KNN showed competitive performance, especially after hyperparameter optimization, but it lagged behind CNN. Decision Tree had the lowest accuracy, as it struggles with high-dimensional data and lacks the feature extraction capability of CNN.

TABLE I Model Results for Arabic and English Datasets.

Language	Model	Accuracy (%)	Training Time (s)
Arabic	Decision Tree	93.27	0.254553
Arabic	KNN	98.05	2.695017
Arabic	CNN	99.15	0.970138
English	Decision Tree	87.84	0.071862
English	KNN	97.05	2.595227
English	CNN	99.07	0.552717

B. *Training Time:*

CNN required the longest training time due to its complexity and large number of parameters. KNN training time was moderate, but its prediction phase is slower compared to Decision Tree. Decision Tree was the fastest to train, making it suitable for applications where speed is prioritized over accuracy.

C. *Comparison Across Languages:*

For Arabic digits, the accuracy of all models was slightly lower compared to English digits, likely due to the complexity and variability in Arabic handwriting styles. The English dataset, being more standardized and widely used, yielded better results.

D. *Impact of Dataset:*

The quality and balance of the datasets significantly influenced the results. The larger size and standardization of the MNIST dataset contributed to its models' higher performance.

Normalization and data augmentation techniques helped improve model robustness.

V. The Model

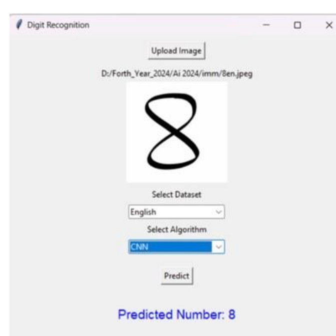

Fig. 1. Handwritten Digit Recognition Application GUI.

The user uploads an image through the GUI. The user selects the dataset (Arabic or English) and the algorithm (Decision Tree, KNN, or CNN). The system preprocesses the image (grayscale conversion, resizing, normalization). The selected model predicts the digit. The predicted digit is displayed to the user.

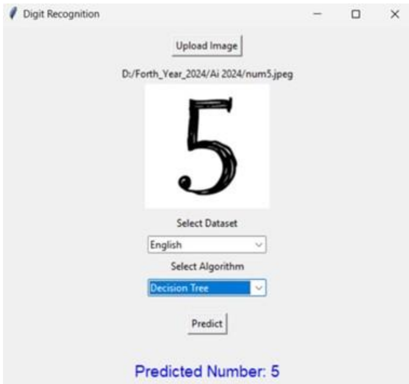

Fig. 2. Number 5 in Decision Tree

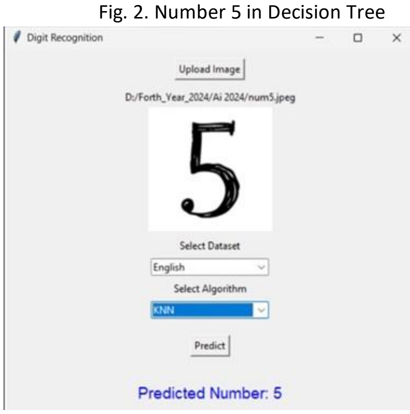

Fig. 3. Number 5 in KNN

Fig. 4. Number 5 in CNN

VI. Confusion Matrices

A confusion matrix is a powerful tool used to evaluate the performance of classification algorithms. It provides a detailed breakdown of a model's predictions compared to the actual labels, enabling us to assess the quality of classification. The matrix is a table with rows representing the actual classes and columns representing the predicted classes.

For this project, confusion matrices were generated for each algorithm (Decision Tree, KNN, and CNN) to analyze their performance on both Arabic and English handwritten digits. The diagonal of the confusion matrix represents the correct predictions for each class. Higher values indicate better preformance for that particular digit, and off-diagonal elements indicate misclassifications. For example, the model might confuse ”1” with ”7” due to their similar shape. Classes with a higher number of misclassifications may indicate imbalance in the dataset or insufficient learning for those classes.

VII. Results

A. Decision Tree:

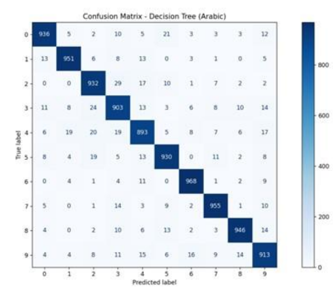

Fig. 5. Confusion MAtrix: Decision Tree(Arabic)

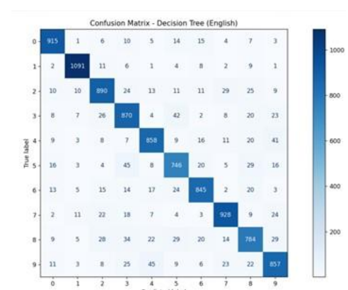

Fig. 6. Confusion MAtrix: Decision Tree

The confusion matrix showed moderate accuracy with some confusion between similar-looking digits, especially in complex Arabic digits.

B. KNN:

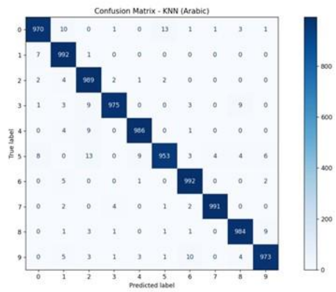

Fig. 7. Confusion MAtrix: KNN(Arabic)

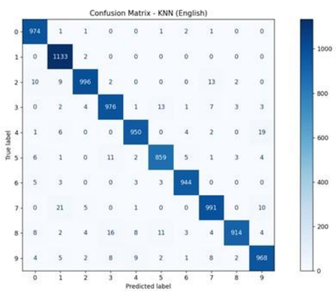

Fig. 8. Confusion MAtrix: KNN

The KNN algorithm performed better than the Decision Tree, with fewer misclassifications. It effectively distinguished between most digits due to the feature-space proximity.

C. CNN:

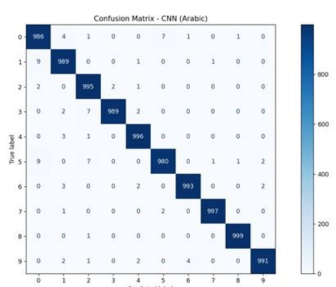

Fig. 9. Confusion MAtrix: CNN(Arabic)

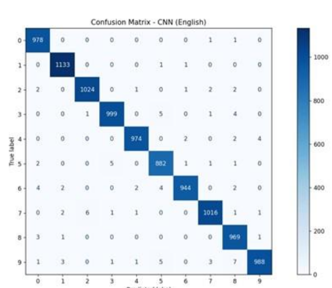

Fig. 10. Confusion MAtrix: CNN

The CNN achieved the highest accuracy, with most predictions falling on the diagonal, indicating correct classification. Misclassifications were minimal, demonstrating its ability to learn complex patterns.

VIII. Conclusion

The project successfully demonstrated the application of machine learning models for recognizing handwritten digits in both Arabic and English. Among the tested models, CNN exhibited superior performance due to its ability to extract hierarchical features, making it particularly effective for complex pattern recognition tasks. While KNN provided competitive results, it requires more computational resources during inference. Decision Tree, though efficient, was less reliable for large datasets. Future enhancements could involve incorporating more diverse datasets and exploring ensemble techniques to improve classification accuracy further

References

[1] https://www.kaggle.com/datasets/mloey1/ahdd1?utm_source=chatgpt.com [Accessed: Jan. 1, 2025].

[2] https://www.kaggle.com/datasets/ahmedshahriarsakib/arabic-digit-dataset . [Accessed: Jan. 1, 2025].