HANDWRITTEN ALPHANUMERIC CHARACTER RECOGNITION USING CONVOLUTIONAL NEURAL NETWORK (CNN)

Dr. A. Gautami Latha[1], Kavya Madhuri. V[2], G. Asha[3], V. Sireesha[4]

Article History: Received: 14.02.2023 **Revised:** 31.03.2023 **Accepted:** 15.05.2023

Abstract:

Recognizing handwritten alphanumeric characters is a complex issue computer vision and pattern recognition fields. However, recent years, usage of deep learning techniques has demonstrated encouraging outcomes in addressing this challenge. Here, we introduce deep learning-based approach which can be effectively used in identifying handwritten alphanumeric characters. Our proposed approach involves pre-processing the handwritten images to extract useful features along with training Convolutional Neural Network (CNN) using the extracted features. We utilize a blend of convolutional layers and fully connected layers to aid the model in learning the representations of the handwritten characters and classify them into their alphanumeric categories. Our approach evaluation is carried on a benchmark dataset of handwritten alphanumeric characters and achieves state-of-the-art results. Our work demonstrates effectiveness of various deep learning techniques for tackling the challenging problem of handwritten alphanumeric character recognition.

Keywords: Artificial Neural Network (ANN), Convolutional Neural Networking (CNN), Deep learning, Handwritten Digit Recognition (HDR), KNN (K Nearest Neighbours), NN (Neural Networks), Object Character Recognition (OCR), Rectified Linear Unit (ReLU), SVM (Support Vector Machine).

[1]Professor, Computer Science and Engineering, Sridevi Women's Engineering College, Hyderabad, India
[2]Computer Science and Engineering, Sridevi Women's Engineering College, B. Tech IV Year, Hyderabad, India
[3]Computer Science and Engineering, Sridevi Women's Engineering College, B. Tech IV Year, Hyderabad, India
[4]Computer Science and Engineering, Sridevi Women's Engineering College, B. Tech IV Year, Hyderabad, India

E-mail address: [1]gauthamilatha2021@gmail.com, [2]kavyamadhuri02@gmail.com, [3]gokaasha35@gmail.com, [4]nayaksirisha06@gmail.com

DOI: 10.31838/ecb/2023.12.s3.338

1. INTRODUCTION

Convolutional Neural Network (CNN) is a deep learning algorithm known for image processing and video processing applications. The spatial hierarchies of features are engineered from input data, which also includes images, in an automatic and adaptive manner.
The key idea behind CNNs is that they perform convolution operations on the input data, which involves kernel - (sliding a small window) or filter on the input data and calculating a dot product between the kernel and the local section of the input data [2]. This operation results in a new feature map, highlighting important patterns and features in the input data. The resulting feature map, is then processed through additional layers of convolution, pooling, and activation functions, which further extract features of the input data. CNNs are commonly used for handwritten character recognition due to their ability to extract and study important features from images, such as edges and shapes, which are essential for recognizing characters [3]. Typically, a CNN is trained on a large dataset of labelled handwritten characters. During training, it recognizes features and patterns of the images that are associated with specific classes (i.e., digits 0-9 and characters a-z, A-Z). Once trained, the CNN can be used to classify new, unseen images of handwritten characters.
Handwritten character recognition uses CNN for automatically recognizing handwritten text or characters and converting them into digital text. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are some of the Deep Learning Methods that have shown great success in solving this problem.
CNN is very effective at tasks like object detection, image classification, image segmentation, and has achieved high level performance on several standard datasets. Speech Recognition and Natural Language Processing are some of its widely known applications.

LITERATURE SURVEY

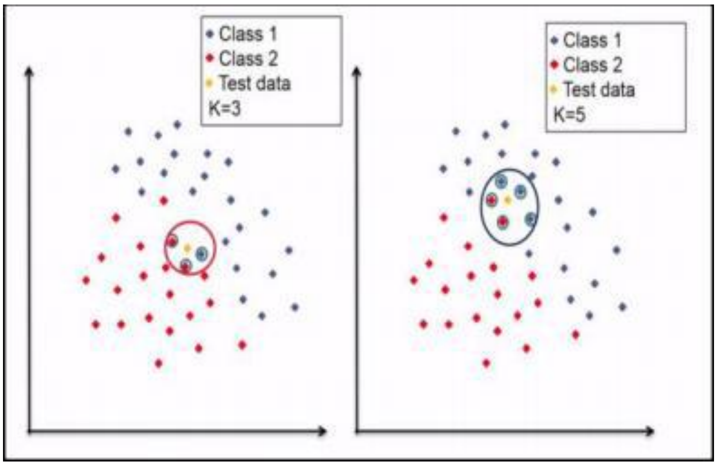
Akanksha Gupta, Ravindra Pratap, et al, “Review on Deep learning Handwritten Digit Recognition using Convolutional Neural Network”, International Journal of Recent Technology and Engineering (IJRTE), Volume - 9, Issue - 5, January 2021:
In this paper, the authors discussed about pre-processing techniques in the digit recognition with coloured documents and complex background with varied intensities.
Hamid, Norhidayu Abdul, et al, “Handwritten recognition using SVM, KNN and neural network”, arXiv preprint arXiv:1702.00723 (2017):
This paper involves the creation of handwritten character recognition model using various algorithms like SVM, KNN and neural networks. At the end of this paper, the authors also describe what method is most likely to be preferred and is more efficient compared to other models.
Reena Bajaj, Lipika Dey, S. Chaudhury, et al “Devanagari Numeral Recognition by Combining Decision of Multiple Connectionist Classifiers”, Vol.27, part. 1, Page No. 59-72, [2002]:
The authors considered various features for classifying Devanagari Numerals. Multi classifier architecture was proposed for increasing the efficiency with 89.6% accuracy.
Anita Pal and Davashankar Singh, “Handwritten English Character Recognition Using Neural Network”, International Journal of Computer Science and Communication, Page No: 141-144, [2011]:
Anita Pal focussed on using neural networks for recognizing English handwritten characters. In this approach, boundary tracing and Fourier Descriptor are used to extract the input features and train the model.

MODELS FOR HANDWRITTEN CHARACTER RECOGNITION

In Handwritten Character Recognition System, algorithms like Support Vector Machine (SVM), Random Forest, K-Nearest Neighbour, and Neural Networks are used for implementation.

K-NN (K-Nearest Neighbour)

The K-Nearest Neighbour is a supervised learning algorithm used in regression and classification problems. The algorithm computes the distance between the new data point and other data points of training dataset using a distance metric like Euclidean distance. It then compares the new data point to the K nearest data points [4]. For classification tasks, the most common among its K nearest neighbours is allotted to the new data point [1].



B. SVM (Support Vector Machine)

Support Vector Machine (SVM) is a popular supervised learning algorithm known for classification, regression, and outlier detection tasks. It identifies the most probable decision boundary (or hyperplane) that splits the data points into various categories [1, 4]. The chosen hyperplane maximizes the boundary between the neighbouring data points of each class. These neighbouring data points are known as support vectors, hence the name "Support Vector Machines". SVMs can work with both linear and non-linear decision boundaries. It can handle high-dimensional data effectively. The parameters of SVM manage the trade-off between fitting the training data and having a smooth decision boundary, which helps to reduce overfitting [5].

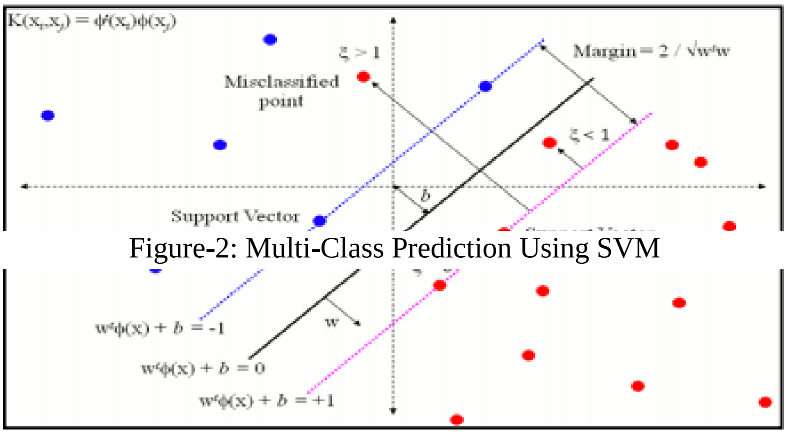


Figure-2: Multi-Class Prediction Using SVM

Neural Networks

Neural network is a deep learning methodology inspired by the working of a human brain. It consists of a large number of interrelated nodes, or "neurons," organized in layers. Each neuron in a layer accepts input signals from neurons of previous layers, and produces an output signal based on an activation function. Image recognition, speech recognition, natural language processing, and predictive analytics are some applications of neural networks.[9]. It learns by adjusting the weights between neurons based on training data, and is capable of recognizing complex patterns and relationships in data [5]. Feedforward neural networks, recurrent neural networks, and convolutional neural networks are different types of neural networks. Each type of network is designed for specific types of tasks and data [1].

DATASET FOR

HANDWRITTEN CHARACTER RECOGNITION

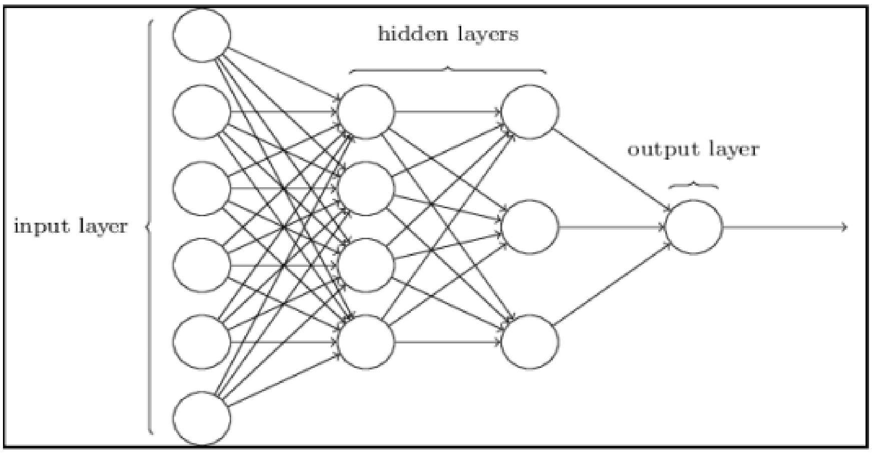


Figure-3: Classification Using NN

The Handwritten Character Recognition system enables the model to predict digits and characters. Two datasets have been used in this project – numeric dataset and character dataset. The overall dataset is of 99.2 MB in size and consists of 22,800 images among which 9,880 images are used for character prediction and 13,000 images are used for numeric prediction.

Numeric Dataset

The Numeric Dataset is an 83.3 MB file containing handwritten digits images to train and test the model for image processing. 10,000 images are considered for training and 3,000 images are considered for testing each with a dimension of 28 X 28 pixels.

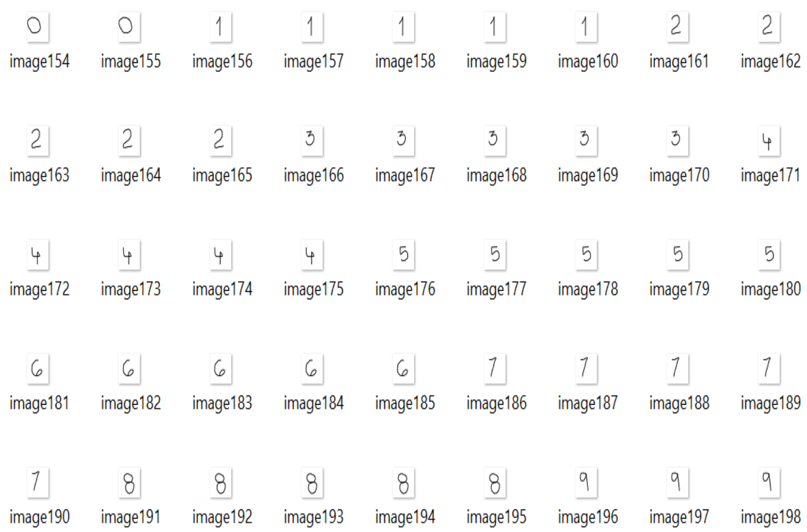


Figure-4: Numeric Dataset

Character Dataset

The Character Dataset is a 15.9 MB file containing handwritten characters images to train and test the model for image processing. 7,800 images are considered for training and 2,080 images are considered for testing each with a dimension of 28 X 28 pixels.

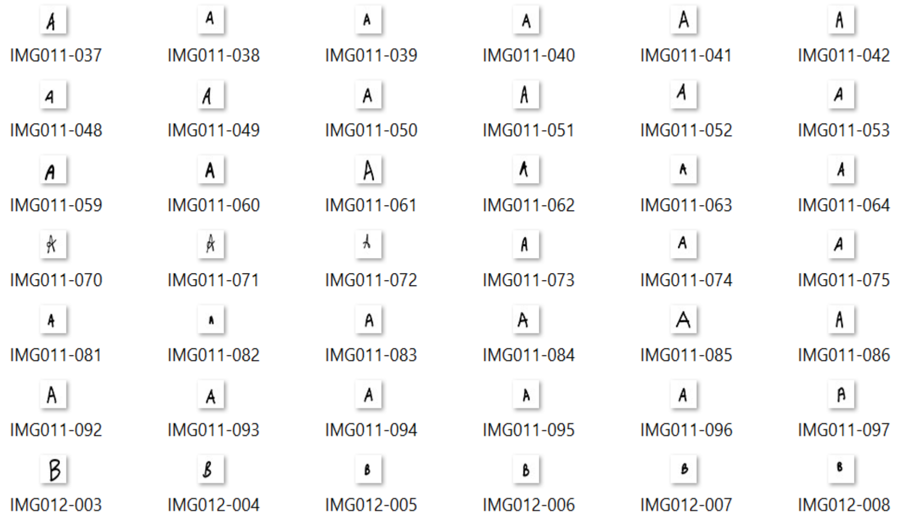


Figure-5: Character Dataset

IMPLEMENTATION OF ALPHANUMERIC CHARACTER RECOGNITION USING CONVOLUTIONAL NEURAL NETWORKING (CNN)

The CNN classifier performs Handwritten Character Recognition. It is a 7-layer convolutional network comprising an input layer of 28 X 28 pixel (784 neurons), followed by five hidden layers and an output layer. The images given as input are greyscale with intensity values varying from 0 to 255, where black signifies 0 and white signifies 255.[1]

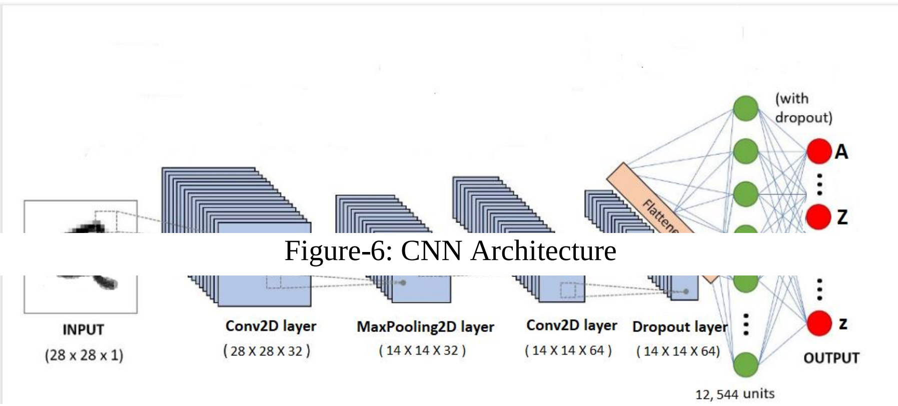


Figure-6: CNN Architecture

The key idea behind CNNs is to use a series of convolutional layers to identify features and patterns in an image [10]. The convolutional layers apply set of filters on input images, which convolutes and produces a set of feature maps for the images, such as edges, textures, or shapes.

A. Layers of CNN

The main layers in a Convolutional Neural Network can be explained as follows:

1) **Input Layer**: This layer consists actual pixel values of the original input image.
2) **Convolutional Layer**: Convolutional layers are responsible for identifying patterns and features in an input image by applying a set of learnable filters. Each filter in a convolutional layer is a small matrix of weights, typically 3x3 or 5x5, that is convolved with the input image to produce a feature map [8]. The convolution operation involves sliding the filter over the entire input image, computing the dot product between the filter and the local input patch at each position, and storing the result in the corresponding position in the feature map.
3) **Pooling Layer**: Pooling layer helps in decreasing the feature map's spatial dimension produced by convolutional layers, while also preserving the important features. This is achieved by replacing each local patch of the feature map with a single value, typically the maximum or average value within that patch. There are several types of pooling layers, including max pooling and average pooling. Max pooling takes the maximum value of each local

patch, while average pooling takes the average value.

4) **Dropout Layer**: The dropout layers are used to arbitrarily "drop out" (i.e., initialize to zero) some portion of the neurons in the network during training. This prevents overfitting by encouraging the model to train on more strong features that are not overly reliant on specific neurons.
5) **Output Layer**: This layer computes the score of classes and outputs that class which has the maximum score

2. RESULTS

Our work presents the following pages – Single Digit Prediction page, Multiple Digit Prediction Page and Character Prediction Page (lowercase and uppercase). This is the title page of our project. It consists of the main heading along with the "About", "Try Your Character", "Try Your Number (Single Digit)", "Try Your Number (Multi Digit)" buttons.

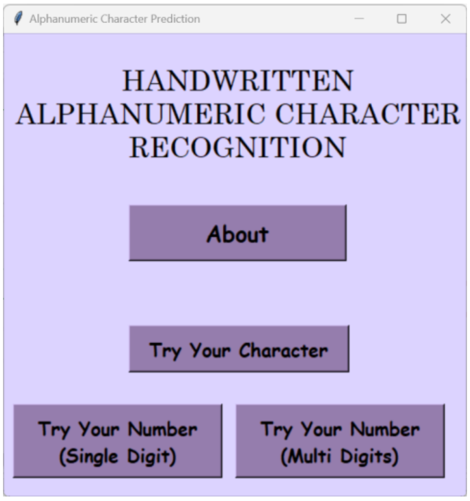


Figure-7: Home Page of Alphanumeric Character Recognition

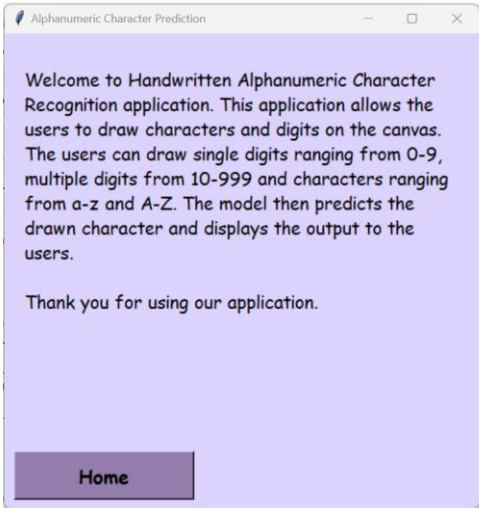


Figure-8: About Page of Alphanumeric Character Recognition

Single Digit Prediction

For predicting single digits, the user must click on the "Try Your Number (Single Digit)" button on the home page. This page consists of a canvas for the user to draw the digit and "Predict" button to display the output, "Clear" button to clear the canvas and "Exit" button to exit the main page.

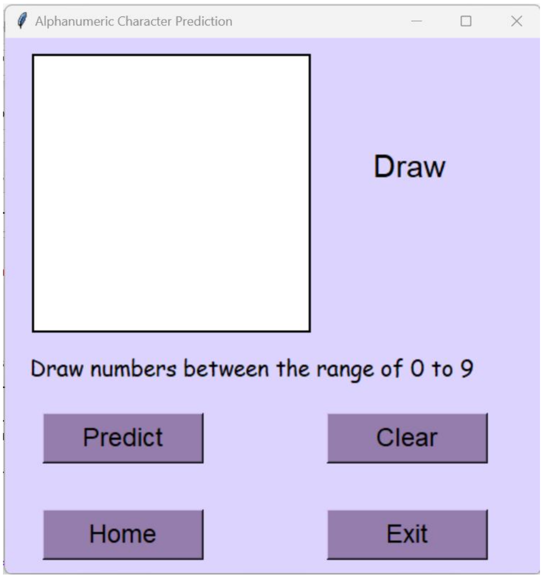


Figure-9: Single Digit Prediction Main Page

The user can draw on the canvas by clicking the left mouse button and dragging the mouse on the canvas. The "Predict" button which when clicked, displays the prediction of the digit drawn on the canvas.

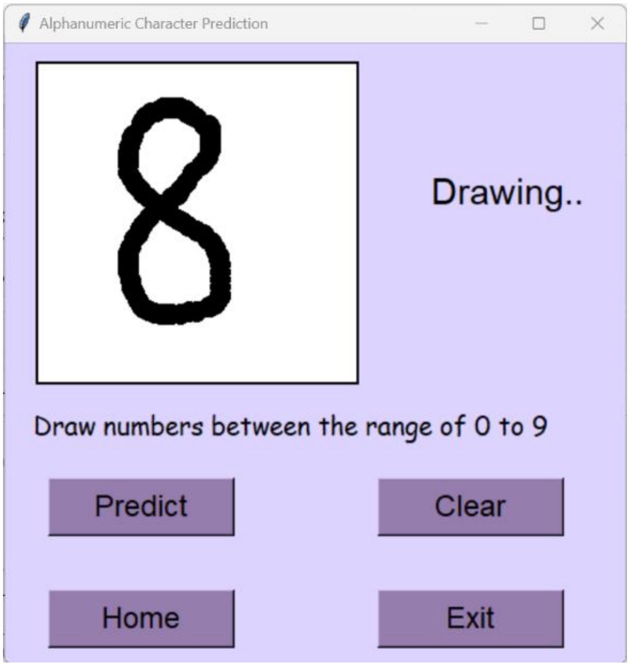


Figure-10: User Draws on Canvas

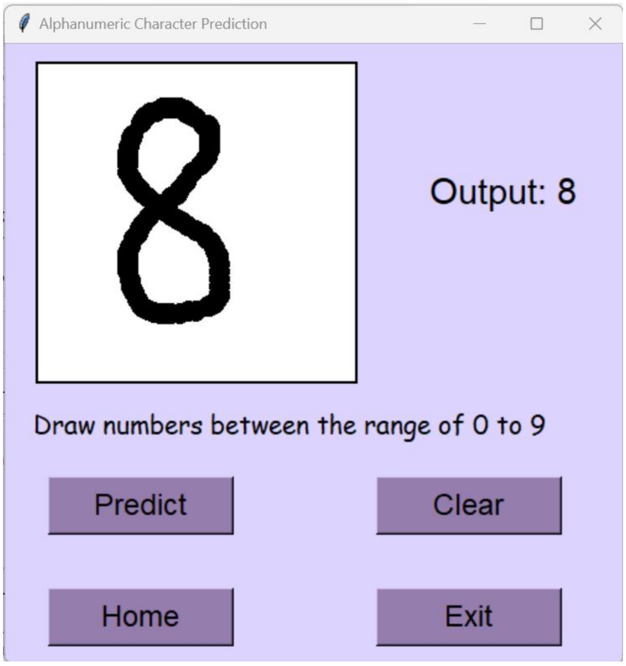


Figure-11: "Predict" Button Displays Output

Multi Digit Prediction

For predicting multiple digits, the user must click on the "Try Your Number (Multi Digit)" button on the home page. This page consists of a canvas for the user to draw the digit and "Predict" button to display the output, "Clear" button to clear the canvas and "Exit" button to exit the main page.

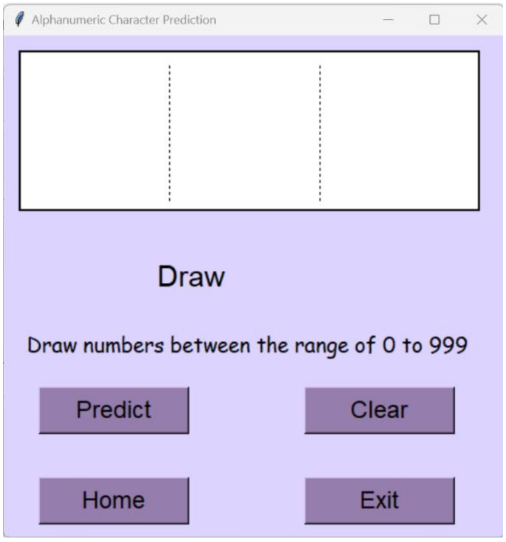


Figure-12: Multiple Digit Prediction Main Page

The user can draw on the canvas by clicking the left mouse button and dragging the mouse on the canvas. The "Predict" button which when clicked, displays the prediction of the digit drawn on the canvas.

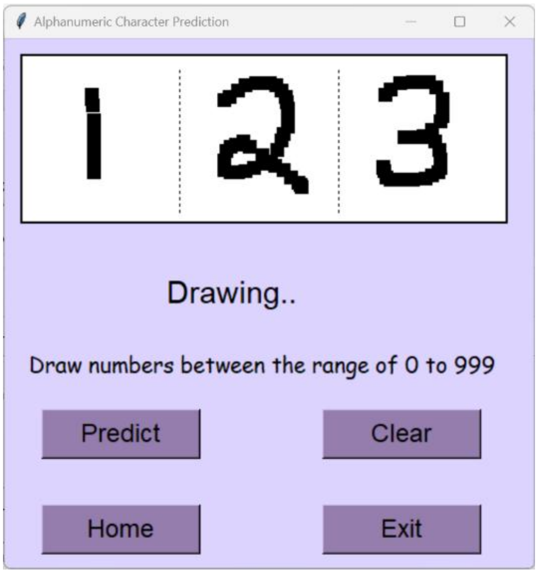


Figure-13: User Draws on Canvas

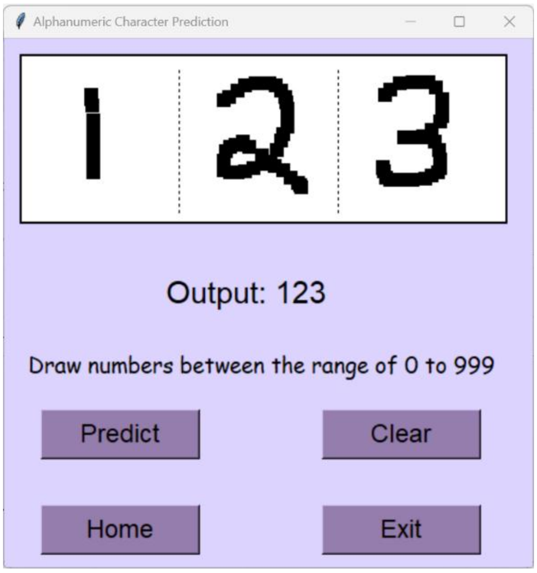


Figure-14: "Predict" Button Displays Output

Character Prediction

For predicting characters, the user must click on the "Try Your Character" button on the home page. This page consists of a canvas for the user to draw the digit and "Predict" button to display the output, "Clear" button to clear the canvas and "Exit" button to exit the main page.

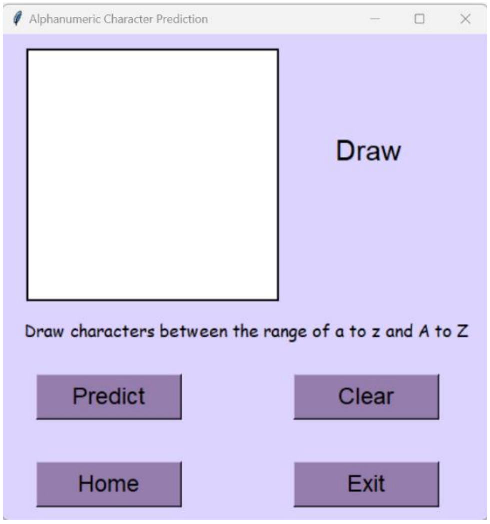


Figure-15: Main Page for Character Prediction

The user can draw on the canvas by clicking the left mouse button and dragging the mouse on the canvas. The "Predict" button which when clicked, displays the prediction of the digit drawn on the canvas. The user clicks on the "Exit" button to view the exit page.

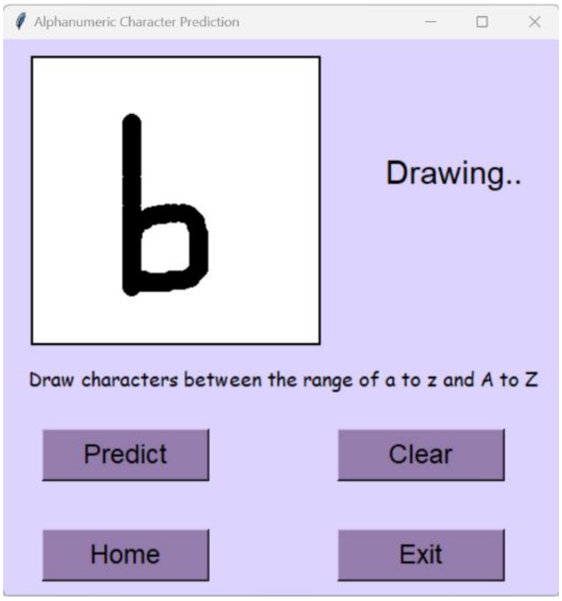


Figure-16: User Draws on Canvas

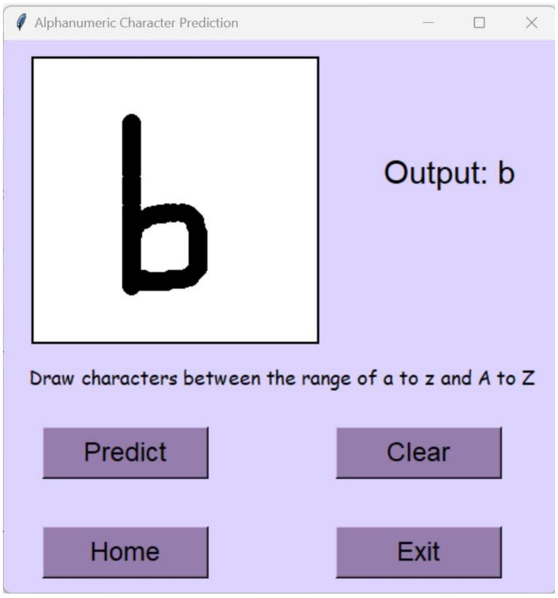
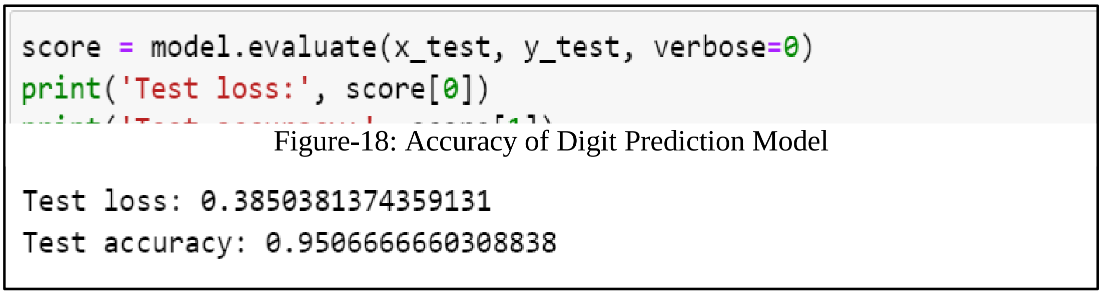


Figure-17: "Predict" Button Displays Output

ANALYSIS

Accuracy and Loss Analysis for Digit Prediction

After running the model for 72 epochs with a batch size of 125, the model was able to achieve test accuracy of 95% and the test loss was calculated to 0.38.



```
score = model.evaluate(x_test, y_test, verbose=0)
print('Test loss:', score[0])

Test loss: 0.3850381374359131
Test accuracy: 0.9506666660308838
```

Figure-18: Accuracy of Digit Prediction Model

The analysis of training and validation accuracy was studied through a graph generated using matplotlib. pyplot library. The graph is as follows:

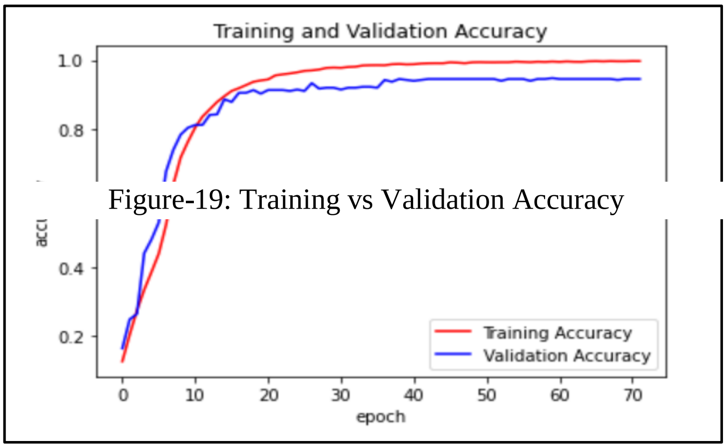


Figure-19: Training vs Validation Accuracy

The analysis of training and validation loss was studied through a graph generated using matplotlib. pyplot library. The graph is as follows:

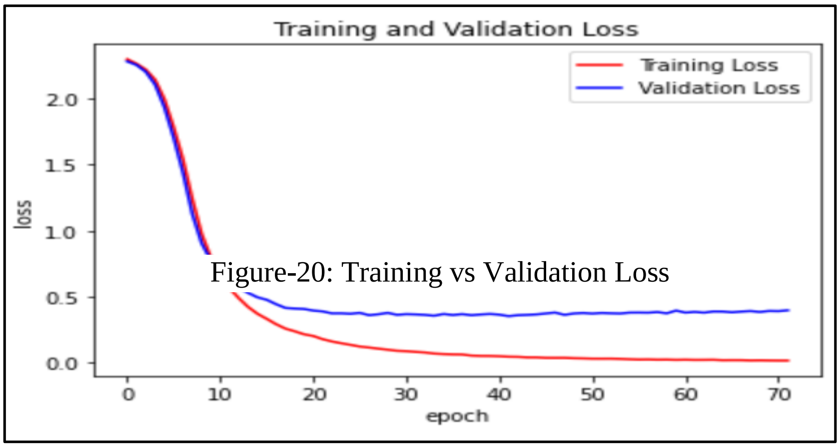
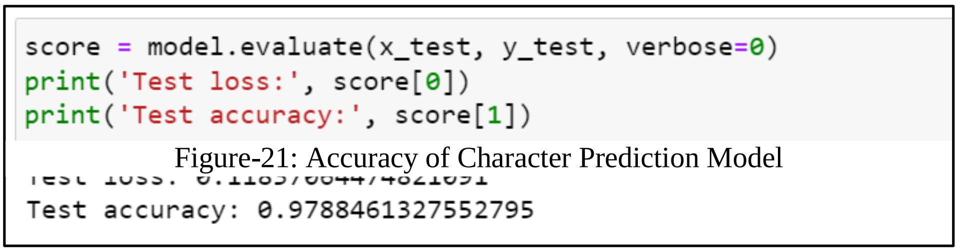


Figure-20: Training vs Validation Loss

Accuracy and Loss Analysis for Character Prediction

After running the model for 95 epochs with a batch size of 62, the model was able to achieve test accuracy of 97.8% and the test loss was calculated to 0.11.



```
score = model.evaluate(x_test, y_test, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])

Test accuracy: 0.9788461327552795
```

Figure-21: Accuracy of Character Prediction Model

The analysis of training and validation accuracy was studied through a graph generated using matplotlib. pyplot library. The graph is as follows:

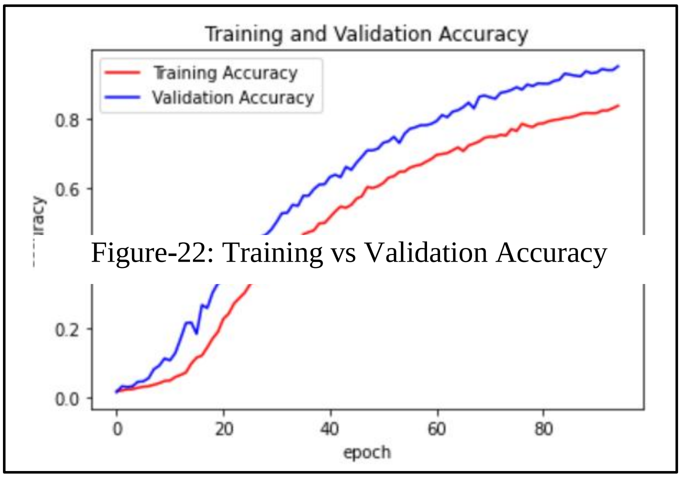


Figure-22: Training vs Validation Accuracy

The analysis of training and validation loss was studied through a graph generated using matplotlib. pyplot library. The graph is as follows:

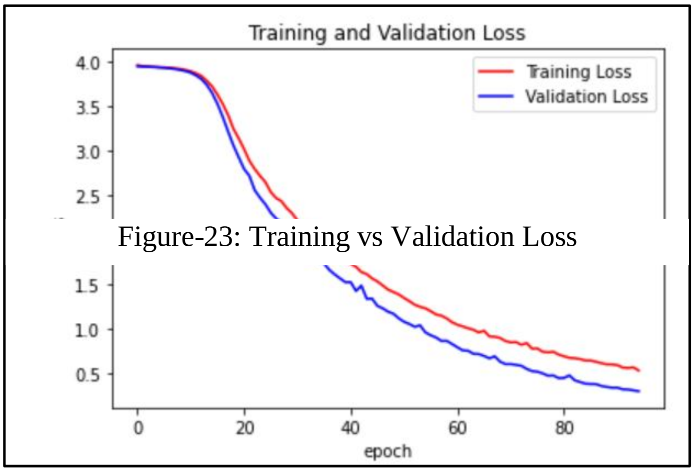


Figure-23: Training vs Validation Loss

3. CONCLUSION

The Handwritten Character Recognition using Deep Learning is an application of the deep learning concept of Convolutional Neural Networks. It consists of a user-friendly Graphical User Interface which enables the users to interact with the system and let the neural network model live predict the digits drawn by user. The Handwritten Character Recognition has wide range of applications in various sectors such as student management, medical, taxation process, banking, etc. In comparison with K-NN, SVM and Random Forest, Convolutional Neural Networking and Deep Learning Approaches are more reliable. As a result, the handwritten digit recognition using Deep Learning is proven to be an effective model with an accuracy of 97% and better efficiency.

4. REFERENCES

[1]. Akanksha Gupta, Ravindra Pratap Narwaria, Madhav Singh, Review on Deep learning Handwritten Digit Recognition using Convolutional

Neural Network, International Journal of Recent Technology and Engineering (IJRTE), Volume - 9, Issue - 5, January 2021.
[2]. Al Maadeed, Somaya, and Abdelaali Hassaine, Automatic prediction of age, gender, and nationality in offline handwriting. EURASIP Journal on Image and Video Processing, no. 1 2014.
[3]. Gaurav Jain, Jason Ko, Handwritten Digits Recognition, Project Report, University of Toronto, 11/21/2008.
[4]. Hamid, Norhidayu Abdul, and NilamNur Amir Sjarif, Handwritten recognition using SVM, KNN and neural network, arXiv preprint arXiv:1702.00723 (2017).
[5]. R. G. Mihalyi, Handwritten digit classification using support vector machines, 2011. [7]. Z. Dan, C. Xu, The Recognition of Handwritten Digits Based on BP Neural Networks and the Implementation on Android, In: 3rd International Conference on Intelligent System Design and Engineering Applications, pp. 1498-1509, 2013.
[6]. Anita Pal and Davashankar Singh, “Handwritten English Character Recognition Using Neural Network”, International Journal of Computer Science and Communication, Page No: 141-144, [2011]
[7]. J. Sadri, C. Y. Suen and T. D. Bui, "A genetic framework using contextual knowledge for segmentation and recognition of handwritten numeral strings", Pattern Recognition, vol. 40, no. 3, pp. 898-919, 2007.
[8]. Reena Bajaj, Lipika Dey, S. Chaudhury, et al “Devanagari Numeral Recognition by Combining Decision of Multiple Connectionist Classifiers”, Vol.27, part. 1, Page No. 59-72, [2002]
[9]. J. Sueiras, V. Ruiz, A. Sanchez and J.F. Velez, "Offline continuous handwriting recognition using sequence to sequence neural networks", Neurocomputing, vol. 289, pp. 119-128, 2018.
[10]. Lee Wells, Shengfeng Chen, Rabia Almamlook and Yuwen Gu, Offline Handwritten Digits Recognition Using Machine learning, 2018.