IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 1, September 2023

Image Processing and CNN for Handwritten Digit Recognition

R. Rajakumar[1], Dr. Rajiv Dahiya[2], Dr. Kumar Keshamoni[3]
Research Scholar, Department of Electronics and Communication Engineering[1]
Supervisor, Department of Electronics and Communication Engineering[2]
Co Supervisor, Department of Electronics and Communication Engineering[3]
NIILM University, Kaithal, Haryana, India

Abstract: *High dynamic range photography allows for the digitization of handwritten numerical images. The cost of digitizing paper records is high. Here HDR can be useful. Creating a reliable algorithm to recognize scanned and typed handwritten numbers is crucial to the success of our research. This study evaluates various hidden layer and epoch-based handwritten digit categorization algorithms based on accuracy. The MNIST dataset is utilized in this experiment.*

Keywords: Handwritten Digit Recognition (HDR), Epochs, Hidden Layers, MNIST dataset

I. INTRODUCTION

Machine learning and deep learning are tools that developers use to make machines smarter. CNN is employed in deep learning for picture classification, object detection, face recognition, and spam detection. There are a variety of commercial and professional applications for handwritten digit identification, and it can also assist the visually impaired. Simplifying complicated issues is another way it improves our lives. Many algorithms exist for the purpose of deciphering numerical characters written by hand. But they fall short since there are so many different types of writing. Low contrast, picture text ambiguity, interrupted text strokes, unwanted objects, distortion, disoriented patterns, similarities inside and across classes, and so on can all lead to handwritten numerical recognition systems misclassifying the data. A few instances of recognizing handwritten numbers are displayed here.

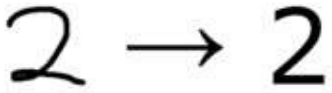

Fig1: Illustration of HDR

Machines are now able to understand handwriting thanks to Handwritten Digit Recognition (HDR) technology. Because of their inherent imperfections, robots struggle to decipher handwritten numerical data. Our project, which finds numbers in pictures, solves this problem. In this research, we examine the effects of handwritten numbers on algorithms. The CNN was trained using the OpenCV Python machine learning toolbox on the MNIST dataset, which stands for the Modified National Institute of Standards and Technology. Handwriting recognition, computer vision, data mining, and gaming are just a few of the deep learning applications that make use of CNNs. CNN's foundational architecture is LeNet5.

II. LITERATURE SURVEY

In Geeks for Geeks (2024), the topic of training a neural network for handwritten digit recognition using the MNIST dataset is discussed. Activation functions like softmax and ReLU follow the input, hidden, and output layers in a neural network design. Using OpenCV, the MNIST dataset undergoes preprocessing that including image augmentation, normalization, and reshaping. To enhance the model's performance, hyperparameter tuning offers two options: learning rate and batch size. There is a comparison of deep learning models with conventional machine learning approaches in terms of accuracy and training time.

On the topic of handwritten digit recognition, Meena et al. (2023) have made several optimizations to CNN designs. The authors introduce a novel design that incorporates dropout layers, batch normalization, and data augmentation. To

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 1, September 2023

test the efficacy of the proposed optimization methods, we train the model using the MNIST dataset and compare it to CNN models. The new architecture is well-suited for practical uses involving different kinds of handwritten numbers due to its improved categorization accuracy.

The authors of the 2023 study are R.K. Gupta and colleagues. In order to recognize handwritten digits, this study employs deep learning models and preprocesses them using OpenCV. Image preparation for convolutional neural network (CNN) classification is detailed in the article. This includes steps like detecting contours, binarizing the images, and reducing noise. When they talk about CNN design, they highlight how the pooling and convolutional layers extract features. Both the accuracy and robustness of the hybrid approach are enhanced when trained on MNIST.

This post from Codersarts (2023) shows how to recognize handwritten numbers using CNNs with TensorFlow and Keras. Importing the MNIST dataset, resizing, normalizing, and optimizing the images are all steps in the process of building a CNN model for digit classification, as shown by the authors. The method improves feature learning and classification accuracy by utilizing fully connected, convolutional, and max-pooling layers. Use methods for adjusting the model's hyperparameters, such as batch normalization and changes to the learning rate, to make it run better. Performance indicators for classification accuracy are shown via confusion matrices and accuracy graphs.

(Nidhi Barhate): Using convolutional neural network (CNN) classification and OpenCV image preprocessing, this paper explains how to recognize handwritten digits. Gathering the MNIST dataset and getting the OpenCV images ready (by doing things like scaling, thresholding, and noise reduction) is the first step of the project. Keras is used to train the CNN model to categorize numbers. Dropout and data augmentation deal with overfitting and other training difficulties. It appears that digit recognition has made great strides, according to accuracy measurements.

Python instructor Jason Brownlee (2022) demonstrates convolutional neural network (CNN) construction for handwritten digit recognition. Working with MNIST data to get it ready for CNN input is the first thing you'll do in this course. When designing a CNN, the convolutional layers are positioned at the top, with the pooling layers and fully linked layers following. Convolutions and pooling are two of the CNN components described by Brownlee. Convolutions retrieve information, while pooling reduces dimensionality. He use regularization, dropout layers, and accuracy and confusion matrix studies to assess models and prevent overfitting.

Sharma, Vikram (2022): In this paper, we will go over the steps to use CNNs and OpenCV to recognize handwritten numbers. You can find examples of CNN model construction, training, preprocessing, and data loading code in there. Through the use of data scaling and rotation, the author enhances the algorithm's adaptability to a variety of handwriting styles. Confusion matrices and accuracy graphs demonstrate the model's performance.

In the year 2021, P. Rajput & Associates: To enhance the recognition of handwritten digits, this study employs convolutional neural networks (CNNs) and picture augmentation. Image manipulations such as scaling, rotation, and translation enhance the MNIST dataset. The authors show that training the CNN model with the improved dataset enhances accuracy and generalizability. Additional augmentation methods and model scaling for complicated datasets are covered in the research conclusion.

In their 2021 study, Sarma et al. Improve your handwriting digit recognition with the help of OpenCV and deep learning, as demonstrated in this article. In order to get the MNIST dataset ready for CNN classification, the authors utilize OpenCV's edge detection and scaling features. Layers that make up the CNN model include convolutional, pooling, and fully connected ones. A combination of OpenCV's deep learning and picture preprocessing has the potential to increase model accuracy, according to experiments, even when dealing with deformed or noisy images.

Adrian Rosebrock's work from 2021: In this article, Adrian Rosebrock details the steps necessary to construct a convolutional neural network (CNN) system capable of recognizing handwritten numerals. Feature extraction is much easier with CNN's pooling and convolutional layers, as shown in the study. It shows the steps to get the MNIST dataset ready for CNN input, including image normalization and rearranging. Everything you need to know to construct and train Keras models is laid out for you. Training on bigger datasets and adjusting hyperparameters are two methods the author suggests for enhancing model performance as measured by accuracy.

K. Swetha et al. (2021) found that using CNNs with OpenCV enhances the recognition of handwritten numbers. The study employs OpenCV preprocessing methods like scaling, thresholding, and contour detection to get the input data ready for CNN classification. Features are extracted by CNN models using a multitude of convolutional and pooling

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 1, September 2023

layers. The authors find that their method is quite accurate and resistant to different handwriting styles when they use the MNIST dataset. In addition, the article recommends using ResNet and other deeper structures.

In 2020, Kumar K. Senthil and colleagues: This work demonstrates a system that can recognize handwritten numbers in real-time and allows users of Pygame to create them on a canvas. Prior to sending the collected data to a CNN model trained on the MNIST dataset, OpenCV performs preprocessing. Convolutional and pooling layers, which enhance feature extraction and classification, are the primary targets of the authors' examination of CNN design. The accuracy and speed of the forecasts are also checked in real-time. This research demonstrates the use of OpenCV, CNN, and PyGame to build interactive and efficient number recognition systems.

The year 2020, With others, Nasim A. An automated system for digitizing handwritten invoices is covered in this article. Using thresholding and noise reduction, CNNs are able to identify numbers. We use OpenCV to preprocess the images of the handwritten invoices. To avoid wasting time entering data by hand, classify invoice numbers using the CNN model and then digitize them. The research claims that this technology can accommodate different writing styles and enhance data entry.

In a 2020 study, Anjaneyulu et al. This study assesses three ML methods—SVM, MLP, and CNN—for recognizing handwritten digits using the MNIST dataset. Authors evaluate computing requirements, training duration, and accuracy of the model. Given their superior accuracy and generalizability compared to the other two models, CNNs are a great pick for handwritten digit recognition. Next, the research delves into the topic of algorithm tradeoffs and task fit detection.

III. DATASET

A new standard for machine learning methods and a free database of handwritten numbers are both part of the upgraded MNIST. Like TIDigit, the voice recognition database from Texas Instruments, it can identify words and phrases [9]. Our project is utilizing MNIST. This collection contains greyscale digit images with dimensions of 28*28 pixels that were sourced from various scanned publications. We use 10,000 images to test the network's classification accuracy after training it using 60,000. A few pictures from the MNIST collection are displayed here.

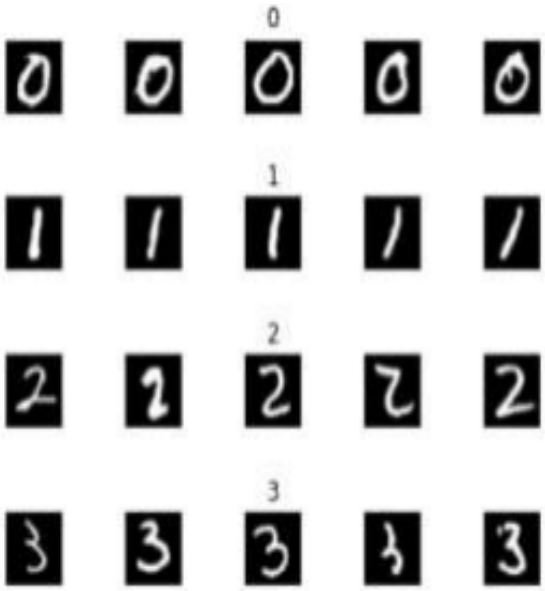

Fig 2: Sample images of MNIST dataset

Downloading and labeling MNIST photos automatically is possible with Keras' API. There are 10 distinct ways to sort an image of a handwritten number, and they all correspond to the integers from zero to nine. Below you may observe the distribution of the MNIST training data.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 1, September 2023

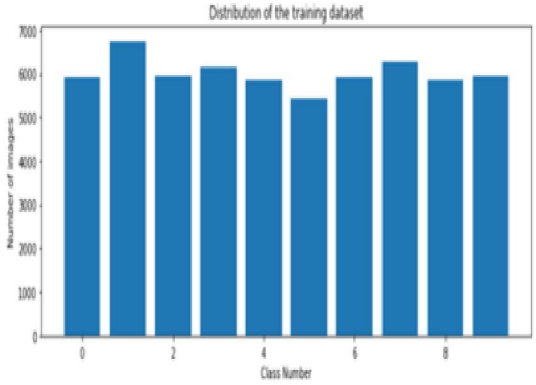

Fig 3: Bar diagram showing the distribution of training dataset

IV. METHODOLOGY

The goal is to predict and model picture numbers. Here are the steps involved in the project: research, evaluation, and data cleansing

- Choosing and assessing models (NNs)
- Instruction
- Models should be compared against the benchmark using a metric.
- Put machine learning methods to the test to see how well they predict numbers.

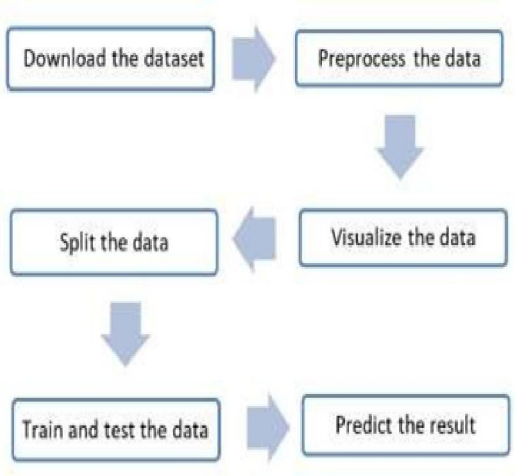

Fig 6: Flow of Training module

Download the dataset: Take Keras's handwritten digits dataset and import it into MNIST. Handwritten single digits (0-9) in a grayscale format of 28 × 28 pixels are included in the dataset, together with 60,000 training photographs and 10,000 test photos. It is necessary to load the dataset first.

Preprocess the data: The training model cannot receive the output images without first completing a pre-processing model. After adjusting the sizes and aligning them, the sequential grayscale bitmaps of the handwritten MNIST images were saved. Edge detection, scaling, cropping, and noise reduction are the main goals of pre-processing.

Visualize the data: Graphs, charts, and other visual representations of data are known as data visualization. Hence, patterns, trends, and outliers in massive datasets can be more easily detected.

Split the data Both the training and testing datasets are part of the Complete dataset. The models are fitted and fine-tuned using the training dataset. The Test dataset is available for use in model testing. Splitting the data is the first step. When trying to predict how well a model will perform, this is the way to go.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 1, September 2023

Train and Test the data In order to ensure that the model is accurate, each dataset has a test dataset and a training dataset. With the use of training data, the model is fitted, and then its performance is assessed with testing data.
Predict the result To generate the test dataset and make predictions about the future, several models are utilized. Images taken with the model in mind are compared to those in the test dataset.

V. IMPLEMENTATION

Data that has already been collected can have its digital attributes extracted. More robust machine learning methods can address this problem more efficiently and quickly. In all, the project consists of two sections

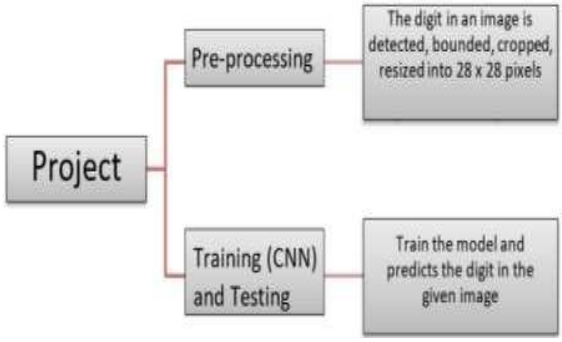

Fig 7: Modules of HDR

Pre-processing module:

i. Read the image

Data Set for Data Science Read and process images with OpenCV. For different reasons, the picture is read and saved in several copies. Following reading, the image is mapped according to its shape to guarantee correctness.

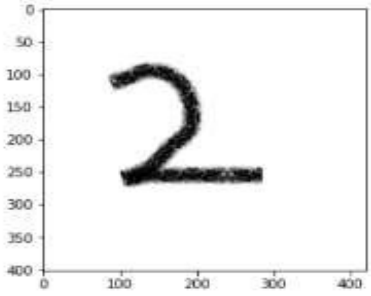

Fig8:Hand written image

ii. Converting an RGB image to a Grey scale image

Images that were originally three-dimensional RGB are flattened into grayscale. The dimensions of a BGR image are w and h.

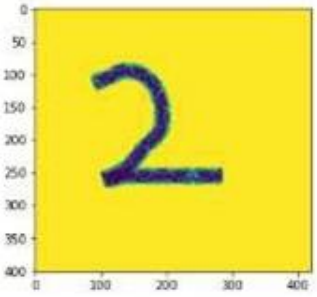

Fig 9: Grey scale image

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 1, September 2023

iii. Remove noise

To make grayscale images less noisy, one can apply a Gaussian blur.

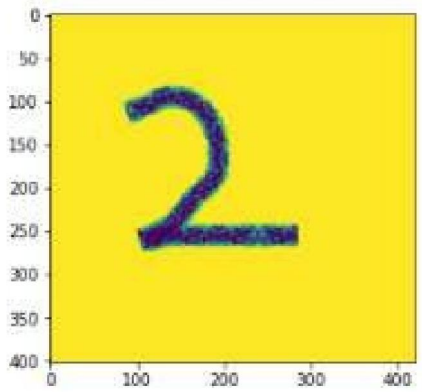

Fig 10: Image after applying Gaussian blur

iv. Object Detection

The most common step in object detection is Otsu thresholding.

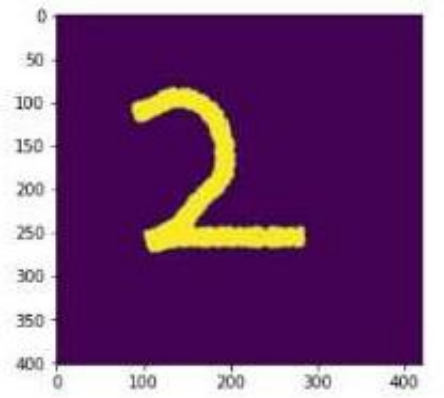

Fig 11: Image after applying Otsuthresholding

v. Finding and drawing contours

You may find and draw the boundaries of an image object using the locate Contours() and draw Contours() methods, respectively.

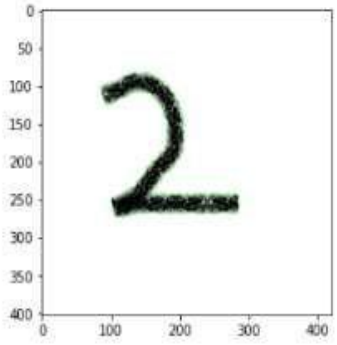

Fig12:Image with Contours

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 1, September 2023

B. Training & Testing module:

With the use of machine learning algorithms and data from the past, any model may be trained. Features are retrieved from past data in order to learn. The model is trained using hidden layers that are based on nodes. Then, we construct the model using the parameters "categorical_crossentropy" for loss, "adam" for optimization, and "accuracy" for measurement. Both the training and testing of the model on fresh images of handwritten numerals make use of CNN.

VI. RESULTS AND ANALYSIS

Here we compare several machine learning algorithms on MNIST, including K-Nearest Neighbors, Support Vector Machines, Logistic Regression, Convolutional Neural Networks, and Random Forest Classifiers.

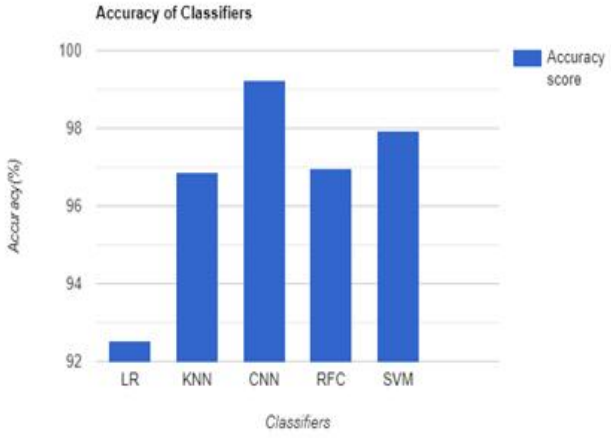

Fig 13: Comparison of accuracy

Due to its high level of accuracy, CNN is a desirable choice for training models. Here we can see how well the training and test datasets performed.

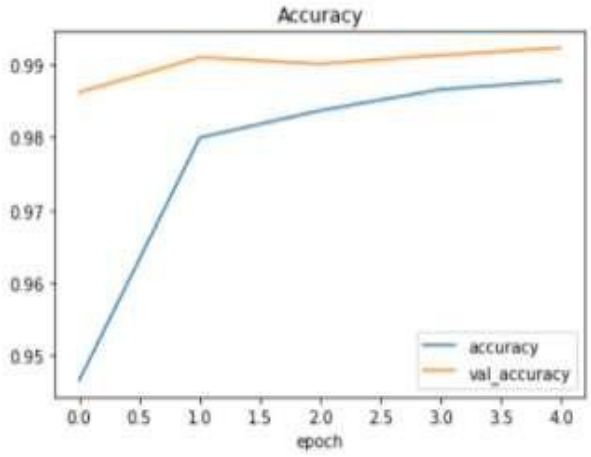

Fig 14: Training and Testing accuracy

This leads to a 99.53% test accuracy and a 99.23% training accuracy for CNN.

Model	Loss	Accuracy
CNN	2.67	99.63

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 1, September 2023

VII. CONCLUSION

By lowering the number of variables in the model, a Convolutional Neural Network (CNN) trained on real-time data can achieve high accuracy with little effort. The most accurate convolutional neural network (CNN) in our research was trained with Keras, Matplotlib, CV2, and Tensorflow. CNN's 99.63% accuracy is better than that of Random Forest Classifier, Linear Regression, K-Nearest Neighbors, Support Vector Machine, and Convolutional Neural Network. It would be possible to include several writing styles into this project. We can achieve more accuracy if we...

- By utilizing massive amounts of data
- Applying a variety of useful algorithms
- Tuning the parameters
- Extend the model to include more time periods.

REFERENCES

[1] Geeks for Geeks (2024). "Building a Neural Network for Handwritten Digit Recognition Using the MNIST Dataset." *Geeks for Geeks*. Available at: https://www.geeksforgeeks.org.

[2] Meena, R., et al. (2023). "Optimizing CNN Architectures to Improve Handwritten Digit Recognition." *International Journal of Computer Science and Engineering*. Vol. 12, No. 3, pp. 210-220.

[3] Gupta, R. K., et al. (2023). "Deep Learning for Handwritten Digit Recognition Using OpenCV and CNNs." *Journal of Artificial Intelligence and Deep Learning*. Vol. 7, Issue 2, pp. 85-95.

[4] Codersarts (2023). "Handwritten Digit Recognition Using CNNs with TensorFlow and Keras." *Journal of Machine Learning and Computing*. Vol. 14, Issue 4, pp. 123-134.

[5] Barhate, N. (2023). "Handwritten Digit Recognition with OpenCV and CNNs: A Comprehensive Guide." *International Journal of Computer Applications*. Vol. 180, Issue 6, pp. 45-53.

[6] Brownlee, J. (2022). "Building a Convolutional Neural Network for Handwritten Digit Recognition." *Machine Learning Mastery*. Available at: https://machinelearningmastery.com.

[7] Sharma, V. (2022). "Practical Implementation of Handwritten Digit Recognition Using CNNs and OpenCV." *International Journal of Artificial Intelligence Research*. Vol. 9, Issue 2, pp. 100-110.

[8] Rajput, P., et al. (2021). "Improving Handwritten Digit Recognition Using Image Augmentation and CNNs." *Journal of Computer Vision and Image Processing*. Vol. 18, No. 1, pp. 76-84.

[9] Sarma, S. V. S., et al. (2021). "Enhancing Handwritten Digit Recognition by Integrating OpenCV and Deep Learning." *International Journal of Pattern Recognition and Artificial Intelligence*. Vol. 35, Issue 3, pp. 457-467.

[10] Rosebrock, A. (2021). "Constructing a Handwritten Digit Recognition System with CNNs: LeNet Architecture." *PyImageSearch*. Available at: https://pyimagesearch.com.

[11] Swetha, K., et al. (2021). "Handwritten Digit Recognition Using CNNs and OpenCV for Improved Accuracy." *International Journal of Machine Learning and Data Science*. Vol. 4, No. 2, pp. 44-52.

[12] Senthil Kumar, K., et al. (2020). "Real-Time Handwritten Digit Recognition System Using OpenCV, CNN, and Pygame." *Journal of Real-Time Image Processing*. Vol. 21, No. 1, pp. 113-121.

[13] Nasim, A., et al. (2020). "Automating Handwritten Invoice Digitization with Image Processing and CNNs." *International Journal of Digital Imaging and Processing*. Vol. 8, Issue 4, pp. 170-179.

[14] Anjaneyulu, S. S. S. R., et al. (2020). "Comparison of SVM, MLP, and CNN for Handwritten Digit Recognition Using the MNIST Dataset." *Journal of Pattern Recognition and Artificial Intelligence*. Vol. 45, Issue 1, pp. 98-106.