

**SEMINAR**

END - TERM PROJECT EVALUATION



# TOPIC

**HAND WRITTEN DIGIT RECOGNIZER**

**SUBMITTED BY: AYUSH GOEL(22CSU040)**

**AARYAN PARGANIHA(22CSU005)**

**ISHAAN POONIA(22CSU086)**

**INDEX**

* **INTRODUCTION**
* **OBJECTIVE**
* **DATASET INFORMATION**
* **NEURAL NETWORK ARCHITECTURE**
* **MODEL TRAINING PIPILINES**
* **MNIST DATA HANDLING**
* **IMAGE PRE-PROCESSING**
* **CONCLUSION**
* **REFERENCES**

## INTRODUCTION

Handwritten digit recognition is an essential problem in computer vision and artificial intelligence. It consists of automatic identification and classification of hand-written digits (0–9), an endeavour that is difficult with substantial variations in individuals' handwriting styles. It is a traditional problem but still an indispensable benchmark for testing and verifying machine learning and deep learning approaches.

The main objective of this project is to develop a stable system that can identify handwritten digits correctly with the help of neural networks. The system is trained on the MNIST database, a popular collection of handwritten digits that consists of 60,000 training images and 10,000 test images. Each image is a 28×28 grayscale image with a digit centered.

For making the system more applicable in real life, preprocessing on images is done using OpenCV in order to pre-process external digit images prior to classification. A feedforward network is built and implemented from scratch in Java without any dependency on high-level machine learning libraries. This will give a clear picture about the internal mechanics of neural networks such as forward propagation, loss computation, backpropagation, and weight updating.

Through the integration of deep learning concepts and image processing methods, this project illustrates a successful method of addressing digit recognition problems with high precision.

## OBJECTIVE

The aim of this project is to create a robust and effective handwritten digit classifier system which can recognize digits from 0 to 9 in gray-scale images automatically. This is done using a feedforward neural network model that is trained on the MNIST dataset, which is a standard against which image classification systems are usually compared.

The project will develop the model from ground up in Java, gaining hands-on experience with the underlying building blocks of neural networks including forward propagation, loss functions, backpropagation, and optimization algorithms like gradient descent. It also involves image preprocessing methods using OpenCV to pre-process real images before classification so that there is consistency in input size and quality.

By the project completion, the aim is to have a high degree of prediction accuracy, proving the model can generalize well to new data. This system provides a basis for more complex applications in optical character recognition (OCR).

## DATASET INFORMATION

### The MNIST dataset, a large dataset of handwritten digits, is the training and test dataset for digit recognizer. It consists of:

### - 60,000 images for training

### - 10,000 images for testing

### - Each image is 28×28 pixels in grayscale

### - Each image has a centered digit ranging from 0 to 9

### This dataset is used for benchmarking image classification models because it is simple yet effective.

A screenshot of a black and white number

AI-generated content may be incorrect.

## NEURAL NETWORK ARCHITECTURE

The neural network design employed in this project is a straightforward yet effective feedforward neural network for classifying handwritten numbers. It has three layers: input layer, hidden layer, and output layer. There is one image per digit from the MNIST dataset, each being 28×28 pixels, and these are flattened into a one-dimensional input vector of 784 values for the network.

The input layer is connected to a hidden layer with 128 neurons. The hidden layer employs the ReLU (Rectified Linear Unit) activation function, which provides non-linearity so that the network can learn complicated patterns in the data. The output layer has 10 neurons corresponding to one of the digit classes (0 through 9) and employs the Softmax activation function to output a probability distribution over all classes, enabling the network to make a prediction.

A computer screen shot of a network

AI-generated content may be incorrect.

## MODEL TRAINING PIPELINE

Training a neural network is comprised of several stages, ranging from propagating the input forward in the network to updating weights by the loss.

The pipeline comprises:

1. Forward Propagation: Computes activations of the output layer by layer.

2. Loss Computation: Quantifies how far off the predicted output is from the true label via Cross-Entropy Loss.

3. Backpropagation: Computes gradients of the loss with respect to weights applying the chain rule.

4. Parameter Update: Updates weights via Gradient Descent optimization algorithm.

Training Configuration:

- Batch Size: 32

- Epochs: 10

- Learning Rate (α): Optimized to achieve good convergence

A diagram of a model training pipeline

AI-generated content may be incorrect.

## MNIST DATA HANDLING

The MNIST dataset is available in IDX binary format. It consists of two primary files:

- train-images.idx3-ubyte: Holds the pixel values of the handwritten digits

- train-labels.idx1-ubyte: Holds the labels of the corresponding digits

The preprocessing operations are:

- Parsing headers to find the number of items and dimensions

- Reading the image data and scaling the pixel values to a range of 0 to 1

- Converting labels to one-hot encoded format appropriate for training the neural network

## IMAGE PRE-PROCESSING

To process actual handwritten digit images, a few preprocessing steps are executed using OpenCV:

1. Read the image in grayscale

2. Resize the image to 28×28 pixels to align with the input shape

3. Use Otsu's thresholding to binarize the image

4. Detect contours to identify the bounding box of the digit

5. Center the digit on a 28×28 canvas

6. Normalize the pixel values to the range [0, 1]

7. Flatten the image into a single-dimensional vector of 784 components

A collage of images of symbols

AI-generated content may be incorrect.

## CONCLUSION

In summary, a strong handwritten digit recognition system was successfully implemented using a neural network trained on the MNIST dataset. The model was written in Java, and image preprocessing was done using OpenCV to maintain consistency and accuracy in input data. Even without the use of high-level machine learning libraries, the model was able to learn well and achieve a high accuracy of 97.85%. This project demonstrates the versatility of neural networks and preprocessing methods in tackling actual-world classification tasks.

**A screenshot of a computer

AI-generated content may be incorrect.**

## REFERENCES

1. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE, 86(11), 2278–2324. https://doi.org/10.1109/5.726791

2. MNIST Database – Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. http://yann.lecun.com/exdb/mnist/

3. OpenCV Documentation – Image Processing and Computer Vision Library. https://docs.opencv.org/

4. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. http://www.deeplearningbook.org

⸻