# MNSIT Handwritten Digit Recognition using CNN

Shafaque Ahmareen
*Computer Science,*
*Emirates School Establishment*
Fujairah, UAE
shafaqueahmareen@ieee.org

Alreem Khalid Khamies Alabdouli
*Emirates School of Establishment*
Fujariah, UAE
stuf2013027132@ese.gov.ae

Sirisha Polturi
*Faculty of Science and Technology*
*(Icfai Tech), Department of Computer Science and Engineering,*
*(of Affiliation)*
*ICFAI Foundation for Higher Education,*
Hyderabad, India-501203
sirisha.polturi@ifheindia.org

*Abstract*— **This research study uses the MNIST dataset to investigate how Convolutional Neural Networks (CNN) might be used to recognize handwritten numbers. From collecting data to evaluating models, it covers it all. We feed the preprocessed MNIST dataset into a CNN. It contains 60,000 training images and 10,000 test images. Using a range of hyperparameters, such as optimizers and learning rates, the proposed CNN model's layers such as convolutional, max pooling, and dense layers are fine-tuned. Additionally, the efficacy of batch normalization method is examined. This study measures the proposed model's performance on test and training data using loss and accuracy metrics to ensure the model can be generalized. This study also proves that CNN works for number recognition and lays the groundwork for improvements in more advanced architectures, data augmentation, transfer learning, and integration with real-time applications.**

***Keywords— MNIST Dataset, Handwritten Digit Recognition, CNN***

## I. INTRODUCTION

With the development of deep learning methods, computer vision has recently seen tremendous progress. CNN has become standard technology, especially for performing image recognition and classification tasks. One of computer vision's most basic and long-standing challenges is using convolutional neural networks to recognize handwritten numbers. [1] Training and testing a CNN model on the MNIST dataset is the focus here because of its status as a well-known benchmark in this field. One of the most important resources for machine learning is the MNIST dataset, which stands for the Modified National Institute of Standards and Technology dataset. Countless handwritten numerals, each represented by a grayscale image of 28x28 pixels, make up this collection. A full foundation for training and assessing ML models is provided by this dataset, which comprises 60,000 training photos and 10,000 test images. The dataset's clarity and simplicity have made it a great place to test new algorithms and approaches for image processing and, more generally, digit recognition. [2] Because of their inherent flexibility and capacity to learn feature hierarchies from input images automatically and adaptively, CNN is well-suited for application in this setting. Deep CNN has changed the game regarding machines' understanding and interpreting visual input. These networks have numerous layers specifically built to identify various elements in images. Classification of the image into a particular category, here, a number from 0 to 9—occurs when deeper layers can detect more complicated elements, such as edges and curves, the basic features normally identified by the initial layers. [3]

The same approach is applied in the context of this research to prepare the images from the MNIST dataset to provide as an input to the CNN model. As part of this preparation, the images are reshaped so that CNN can read them correctly, and the pixel values are normalized to a uniform range. [4] This phase is vital to ensure the model can learn from the best available input data. After data preparation, the following step is to create the CNN model. Layer kinds and numbers, activation function usage, and other architectural decisions are all part of this process. There is a distinct function for each CNN layer. For instance, to generate feature maps that emphasize image features, convolutional layers subject the input to a battery of filters. When these feature maps are pooled, their dimensionality is reduced, simplifying the computations needed and helping minimize overfitting. The output image's classification as a number between 0 and 9, is produced by fully linked layers. During training, the CNN is fed with the MNIST dataset's images, and its weights are adjusted iteratively to minimize the discrepancy between the predicted and actual numbers. Metrics such as accuracy and loss track the training process and provide insights into the model's learning performance. This study has examined the optimization methods with varying learning rates to identify the best combination. These algorithms include Stochastic Gradient Descent (SGD), Adam, and Adadelta. The training step is just the beginning of the evaluation process for the model's performance. Using the MNIST dataset's test set, which consists of unseen data. An essential component of any machine learning model, this step shows how well it generalizes, making it vital. We examine the test set's performance using the same accuracy and loss metrics used during training. [5]

An existing research study produces an accurate model for handwritten digit identification and sheds light on how different configurations and methodologies work with CNN for digit recognition. [6] These discoveries include understanding the effects of different preprocessing approaches on the model's learning capacity, the impact of different architectures and hyperparameters on performance, and how additional techniques, such as batch normalization, can improve the model's capabilities even more.

This research study proves that CNNs are the real deal regarding image identification. It shows that even a basic dataset like the MNIST can teach CNN a lot and get quite accurate results regarding digit recognition. In addition to significantly impacting computer vision, this project's techniques and results open the door to future studies and

applications that use convolutional neural networks CNN for various image processing and recognition jobs.

### A. Aims and Objectives

The main objective of this study is to show that using the MNIST dataset as a testbed is successful for handwritten digit recognition. One of the goals is to create a CNN model that can correctly assign a number between zero and nine to each image. Optimal CNN performance requires careful dataset preparation, an efficient network design, and hyperparameter selection. The effect of various optimization algorithms and sophisticated approaches, such as batch normalization, on the model's performance should also be investigated and measured. In addition to improving digit classification accuracy, this study aims to learn what makes CNN good at image recognition, advanced computer vision, and machine learning.

We have used CNN throughout this project. The CNN algorithm works very well with image-related tasks and helps achieve high accuracy and sustainable results. This is the reason why this project has used CNN algorithm.

### B. Research Questions

- When using the MNIST dataset for handwritten digit identification, how do the number and type of layers in a CNN influence efficiency and accuracy?
- How do various optimization methods and their learning rates affect CNN's ability to recognize handwritten numbers? Some examples of these algorithms are Adam, SGD, and Adadelta.
- On the MNIST dataset, how can normalization and batch normalization, two preprocessing approaches, affect a CNN model's learning capacity and overall classification accuracy?

### C. Dataset's Description

Machine learning researchers and practitioners rely on the massive MNIST dataset, an abbreviation for "Modified National Institute of Standards and Technology," to train and evaluate their models. The total number of images is 70,000, with 60,000 serving as training examples and 10,000 as test examples. [7] Each grayscale bitmap image in the collection represents a single integer from 0 to 9, which has dimensions of 28x28 pixels. The images labeled with the correct digit provide a simple and unambiguous structure for supervised learning tasks. This dataset is widely used as a benchmark in computer vision, particularly for image identification and classification tasks, due to its size and notoriety for simplicity.

Generally, the dataset is all about images. All images of the dataset contain different versions of numbers from 1 to 9. The size of the dataset is large, which helps to create the model with higher accuracy.

### D. Ethical Issues

Although mostly technical, several ethical concerns about using CNN for digit recognition on the MNIST dataset are raised. Data privacy and permission concerns are paramount, particularly as the research moves beyond MNIST and into real-world data that could include personally identifiable information. It is of the utmost importance to always obtain clear consent before processing any new data and to adhere to data protection requirements. Potential bias in ML models is another area of concern from an ethical standpoint. Even though the MNIST dataset is simpler and less biased than others, the established methodology could be used on more complicated datasets in the future, possibly unintentionally increasing biases. [8] This calls for thoughtful deliberation and frequent evaluation to guarantee that the model's forecasts are objective and fair.

The rest of the paper is divided into the following parts.

- In section II, there is a detailed literature review, along with some important terminologies of CNN
- In section III, the methodology used to complete this research project is discussed.
- In section IV, analysis has been discussed along with their respective results.
- In section V, there is a conclusion that concludes this research project. Future recommendations are also given in this section.

## II. LITERATURE REVIEW

An improved model for MNIST handwritten digit classification using deep CNN is introduced in this paper. Using hyperparameter optimization is crucial for the model to work. Three feature extraction layers, activation and convolution, and two classification layers, density, comprise the CNN architecture. Hyperparameters are adjusted to enhance performance, including learning rate, activation function, batch normalization, kernel sizes, and batch sizes. [9] The model's average classification accuracy was 99.4 percent on the test dataset and 99.82% on the training dataset, which is rather impressive. The authors say you need to optimize your hyperparameters to get good results. They go over how batch size affects training results, drawing attention to the need to strike a compromise between memory efficiency and model convergence. Another important part is learning rate decay, which helps the model converge faster by gradually decreasing the learning rate during training. This method improves the model's stability and helps prevent local minima. Techniques such as dropout layers and early halting based on validation loss help deal with overfitting, a prevalent problem in deep learning. Experiments were carried out using 60,000 training and 10,000 testing samples utilizing the MNIST dataset. The authors extensively researched the effects of various hyperparameters on the model's output. Compared to other top-tier models, the model demonstrates even more effectiveness by outperforming them in classification accuracy. A thorough investigation of the effects of hyperparameters in CNN models for handwritten digit recognition, which contributes to the area, is provided by this study. Impressive accuracy rates on the training and testing datasets demonstrate how well the model holds up and how it could be used in real-world situations.

This paper [10] uses the MNIST dataset to examine how well three different ML algorithms, Support Vector Machines, Multilayer Perceptron, and CNN, recognize handwritten digits. The study's main objective is to find the best model for digit recognition tasks by comparing their

performance in terms of execution time and accuracy. Starting with the fact that various people have varied writing styles, the authors admit that handwritten digit detection isn't an easy task. They stress the significance of precision in practical contexts, where mistakes can have far-reaching consequences, such as automated bank check processing. The publication describes each algorithm in detail in its methodology section, including the dataset, number of epochs, algorithm complexity, accuracy, and runtime. The MNIST dataset, which includes 70,000 photographs of handwritten digits, is utilized for this assessment. Each algorithm's implementation is described in detail in the paper. While support vector machines (SVMs) are known for their speed and simplicity, they aren't great at identifying complicated images. CNN and MLP, in contrast, can accomplish more sophisticated structures, leading to improved accuracy in number identification. While support vector machines (SVMs) perform best on training data, CNN beat all other models on the test dataset. A crucial component of machine learning models, CNN's better capacity to generalize from training data to unknown data is shown here. We then go over how long it takes for each model to run, comparing how quickly and efficiently SVM works with CNN and how computationally heavy it is. In terms of accuracy, CNN is the best model for handwritten digit recognition tasks, according to this research. Execution times will be longer because of this. Finding the optimal solution for a given job often requires balancing two competing priorities: accuracy and efficiency. Finding the right algorithm, one that strikes a balance between accuracy, execution time, and processing resources, depends on the unique needs of each job, according to the article. CNN is ideal for complicated prediction problems with image data.

With an emphasis on the MNIST and EMNIST datasets, the article offers a thorough overview of developments in handwritten character recognition. New computer vision methods, particularly CNNs, rely heavily on the MNIST dataset, which the study recognizes as crucial for their validation. It traces the history of MNIST as a standard from its inception in 1998 to its current iteration in 2019. This study classifies methods for achieving high accuracy on the MNIST dataset according to whether they use convolutional neural networks, data augmentation, or preprocessing. [11] According to the research, the MNIST dataset is becoming less challenging as state-of-the-art methods have reached test error rates below 1%. In response, a more difficult benchmark known as the EMNIST dataset was introduced in 2017 with a bigger sample size and handwritten letters. Using the EMNIST dataset, the article compares the results of several models, including CNN, capsule layers, and deep convolutional extreme learning machines. It emphasizes the substantial accuracy attained with these models, particularly when utilizing CNNs with sophisticated methods like capsule layers. The authors note that data augmentation approaches frequently improve models' performance on these datasets, especially when combined with CNN classifiers. They also point out that, with advancements in hardware, the sector is moving towards more complicated designs and has a greater ability to manage massive amounts of information. The research concludes that although MNIST and EMNIST accuracy rates are getting close to their maximum, solving more complicated computer vision problems may require new and improved CNNs and other techniques.

The creation of a CNN model for the MNIST dataset-based recognition of handwritten digits is the primary emphasis of this work. The model comprises a fully connected layer, two pooling layers, an input layer, two convolutional layers, and an output layer. Feature extraction relies on the 5x5 convolutional core of the convolutional layers, which, in turn, simplifies the computational process. [12] The 2x2 pooling layers help reduce image resolution. The fully linked layer is employed to decipher these characteristics and generate predictions. The paper's focus on training and the model's performance is crucial. Following rigorous training, the model successfully recognized handwritten numbers with an accuracy rate of 99.25% on the test set and a flawless accuracy rate of 100% on the training set. The impressive level of accuracy demonstrated by the model highlights its promising capabilities for real-world digit identification tasks. Additionally, the authors address the potential drawbacks of their methodology, namely when it comes to classifying digits that are either handwritten or downloaded from the internet. They imply that to enhance the model's capacity for generalization, future studies should look at training it using handwritten custom digital data. To make CNN models more applicable, continuously improving and adjusting them to different datasets is necessary.

Finding digits written by hand using the MNIST database and various machine-learning models is the main topic of this paper. [13] The authors examine options such as Support Vector Machine (SVM), Decision Tree, Naïve Bayes, K-Nearest Neighbor, and Random Forest to evaluate how well multiple algorithms perform in handwritten digit recognition. Improving the accuracy and consistency of number recognition is the main goal of the research. Testing how well and quickly these algorithms work is an important part of the study. For this purpose, the authors put their models to the test using the MNIST dataset, which includes 28,000 images of handwritten numbers for training and 14,000 photos for testing. According to the study, the SVM classifier performed best among the classifiers tested, with a success rate of 95.88%. Although the time it takes to compute the results varies, this finding demonstrates that SVM is effective at correctly classifying handwritten digits. Factors such as human handwriting variability and the necessity for reliable feature extraction techniques are discussed in the article as obstacles to handwritten digit recognition. The authors stress the need to tailor one's choice of machine learning algorithm to one's job, striking a balance between accuracy, runtime, and available computing resources.

With an emphasis on using CNN and K-nearest neighbors (KNN) algorithms, the study offers a perceptive investigation into handwritten digit recognition. To build a strong structure that can accurately detect different types of handwritten numbers, the study uses the MNIST dataset to train and evaluate these models. An important aspect of the study is contrasting the two networks' capabilities in identifying handwritten numbers using CNN and KNN. The CNN model's deep learning framework stands out because it processes and identifies the numerical images well. The convolutional, pooling, and fully connected layers that make up this model are all essential for extracting features and categorizing digits. The capacity of the well-known and straightforward KNN method to categorize numbers according to the closeness of feature space data points is also tested. The author [14] highlights the difficulties of

handwritten digit recognition, including the fact that handwriting styles can vary greatly and the requirement for models to generalize from previously encountered training data to new, unfamiliar data. The study shows that CNNs can handle the intricacy of handwritten digit images better than KNN despite KNN providing a simpler technique. This is due to CNN's deep learning capabilities. The paper goes even further into the nuts and bolts of implementing these models, covering topics like the complexities of training neural networks and the significance of preprocessing datasets. Although both CNN and k-nearest neighbors (KNNs) have their uses, the study finds that CNNs are often better at recognizing handwritten digits despite their higher computational complexity.

The authors [15] offer a CNN-based method for improved handwritten digit recognition. By training and testing the MNIST dataset, the authors aim to improve accuracy while decreasing computational time. This study presents a CNN architecture enhanced with the DL4J (Deep Learning for Java) toolkit to improve the numerical recognition procedure. The research highlights the CNN design with several pooling, fully connected, and convolutional layers. This architecture is crucial for feature extraction and classification from images of handwritten digits. The model uses transformations like convolution and pooling to process input data efficiently, as seen in the implementation details. The research stands out for thoroughly testing the suggested CNN model to confirm its efficacy. The authors reported an impressive 99.21% accuracy rate in detecting handwritten digits, marking a notable advance over earlier proposed models. Optimal design for digit recognition is discussed in the research, which also investigates how changing the number of convolutional layers affects accuracy and error rates. The study highlights the need to balance precision with computing economy. Important for real-world applications, the authors provide an effective training and prediction procedure using the DL4J framework. The study says the model might be enhanced for more complicated recognition tasks, like letter recognition.

This research uses the MNIST dataset to investigate the feasibility of training a CNN to recognize handwritten digits. Improving CNN's accuracy and efficiency in digit identification tasks is the main emphasis of the work. [16] The study presents a novel CNN architecture for feature extraction and digit image classification, incorporating multiple convolutional, pooling, and fully connected layers. With a 99.21% success rate on the MNIST test dataset, the CNN model developed in this work is successful. This impressive precision hints at the model's resilience and possible use in real-world digit identification tasks. To achieve such a high level of accuracy, the model's training procedure is detailed in the paper, emphasizing the significance of layer setup and parameter optimization. In addition, the study delves into the difficulties of using the model on handwritten numbers obtained from different sources, like manual writing or online downloads. The authors propose that future studies use a wider variety of digital handwriting data in training to improve the model's generalization ability. By shedding light on how to create a CNN model that is both efficient and accurate, this study contributes significantly to the area of number recognition. An all-encompassing method for enhancing CNN-based number recognition systems is provided by integrating sophisticated layer designs and focusing on parameter optimization.

Utilizing the MNIST dataset in particular, the article offers a thorough investigation into the application of CNN for recognizing handwritten digits. An adaptive framework for optical character recognition (OCR) models is being developed as part of this project to improve accuracy for specific datasets of numerical digits in English. A fresh bespoke dataset of handwritten digits was created by a single contributor, guaranteeing stylistic consistency; this is one of the study's main contributions. This dataset is used with the conventional MNIST dataset when training a CNN model. The article investigates how different amounts of bespoke data, with and without rotations, impact the model's performance. According to the research, custom data can greatly enhance OCR accuracy for personalized or localized handwritten text. The study also emphasizes the difficulties linked to data imbalance and scarcity in optical character recognition. According to the authors, one possible answer to these problems is to generate data using traditional ways. [17] Considering the limits of existing datasets, they contend that this method might be the gold standard for bespoke data augmentation.

The research uses the MNIST dataset to investigate CNN-based handwriting digit recognition. Feature extraction and digit image categorization are two of the many important tasks performed by the novel CNN architecture it introduces. [18] The model demonstrates phenomenal success with a remarkable accuracy rate of 97.78% on the MNIST test dataset, highlighting its usefulness and promise for real-world applications. Examining the steps used to train the model, the study finds that optimizing parameters and layer configuration is crucial for good results. The article also delves into the difficulties of deciphering handwritten numbers from several mediums, including manual writing and online downloads. The research recommends training the model with a wider variety of handwritten digital data to enhance its generalizability. The study advances digit recognition by suggesting an effective and precise CNN model. It provides a holistic strategy for improving digit recognition systems that rely on CNN by providing insights into sophisticated layer topologies and optimizing parameters.

### A. CNN

CNN is fundamental to deep learning, particularly for processing and analyzing images. They are very good at visual identification, image categorization, and object detection because of their special architecture based on the human visual cortex. Central to CNN are layers, the purpose of which is to process input data, usually an image. Mid to the system is the convolutional layer, which generates feature maps by applying filters to input data. These filters automatically focus on important elements, such as edges or textures. The next layer is the pooling layer, which controls overfitting by lowering the network's computation and parameter counts and shrinking the representation's spatial size. Near the very end of a CNN design are the fully linked layers. This is where the abstract reasoning that relies on the traits that have been extracted happens. A fully connected layer integrates the learned characteristics for the final

classification or prediction job by combining neurons in that layer to all activations in the previous layer.

When it comes to computer vision, CNN has proven game-changers. Applications like medical image analysis, autonomous vehicle navigation, and face recognition benefit greatly from their capacity to learn hierarchical feature representations. CNN has always led deep learning research, and this trend will only accelerate as technology improves and CNN can process increasingly complicated tasks and data kinds. [19]

### B. SGD

Machine learning and deep learning rely heavily on SGD, an essential optimization technique while training their models. SGD is a technique for reducing a training-set-averaged objective function, usually a loss function. Instead of using the whole dataset to calculate the loss function's gradient, as is done by classic gradient descent methods, SGD adjusts the model's parameters using a small number of training samples. For big datasets, this method makes SGD far more efficient. [20]

### C. Adam Optimizer

Regarding deep learning, the Adam optimizer is one of the most popular algorithms for training neural networks. Its ability to handle massive datasets and complex models results efficiently and powerfully from its successful usage of adaptive learning rates. Adam integrates the best features of the RMSProp and AdaGrad optimizers to deal with sparse gradients on noisy issues. For every parameter, Adam keeps track of two moving averages: one for the slopes (like momentum) and another for the square of the gradients (like RMSprop). Using these moving averages, we may determine a parameter-specific adaptive learning rate. First, the mean, which considers previous angles, aids momentum; second, the uncentered variance, which scales the learning rate inversely proportional to the size of the slopes, is very useful. Adam is resilient to scaling and updating differences because this technique dynamically adapts the learning rate for each parameter. [21]

### D. Adadelta Optimizer

A popular machine learning technique for training neural networks, Adagrad has some limitations; the Adadelta optimization algorithm aims to fix these issues and make Adagrad even better. Although Adagrad performed well with sparse data, it frequently encountered problems during training, such as a quickly dropping learning rate, which resulted in less-than-ideal convergence. Adadelta, which provides a more stable method for adaptive learning rates, arose to address these concerns. Reduced aggressive monotonically declining learning rate is the main difference between Adagrad and Adadelta. Adadelta limits the window of accumulated past gradients to a specified size instead of adding all past squared angles. This is accomplished by utilizing exponentially moving averages, which ignore long-gone gradients in favor of more current data. The learning rate can be kept from dropping to very low values with the help of this method. [22]

One area where Adadelta has greatly appreciated is its use with recurrent neural networks (RNNs) and other deep learning applications. For complicated and large-scale ML jobs, it's a good option because it can dynamically change learning rates and consider the most relevant gradient information. Even if there are more recent optimization algorithms, Adadelta is still a useful tool for optimization, particularly for jobs where keeping and changing learning rates are tough. [23]

## III. METHODOLOGY

Data preparation is the project methodology's first and most important phase. The MNIST dataset, which is 28x28 pixels and contains 60,000 training and 10,000 test images of grayscale handwritten digits, is the basis of this work. These images must first undergo several preprocessing stages for machine learning to work. To ensure the data is consistent, we normalize the image pixel values to all fall on a scale from 0 to 1. The images are also prepared for use with CNN by being transformed into a tensor format. This transformation is critical because CNN works best with data that follows a certain format. Building the CNN model follows data preparation. The project's deep learning method revolves around this model. The CNN architecture of this project is organized into numerous levels, with each layer performing a specific task. The convolutional layers are set up with different numbers of filters; they are responsible for extracting features from the input images. To help reduce computational complexity and overfitting, these layers are alternated with max-pooling layers, which reduce dimensionality. Important for the last categorization judgment, the network has thick layers towards the end. Activation functions such as ReLU (Rectified Linear Unit) contribute to the model's ability to learn complicated patterns by introducing non-linearity.

Training the model follows the definition of the model's design. The next step is to feed CNN the training data that has been prepared. The training phase changes the network's weights to minimize the discrepancy between the actual and predicted values. To ensure the model is learning well, watching how it performs as it is being trained is crucial. Common performance measures used for this purpose include loss and accuracy. To find the optimal configuration for the model's performance, we experimented with several optimization techniques, such as Stochastic Gradient Descent (SGD), Adam, and Adadelta, each with its learning rate.

A distinct test set is used to assess the model's performance in addition to the training set. This phase must ensure the model works properly with fresh, unknown data. The accuracy of the model's digit classification is tested at this stage. Some criteria for evaluating performance include loss and precision on the test set. It is possible to learn which optimizer configuration produces the most accurate and error-minimizing outcomes by comparing these metrics across several setups. Moreover, the study investigates the utilization of methods such as batch normalization in conjunction with various optimizers and learning rates. The training process can be made more stable and faster by applying batch normalization to every layer or activation input. The impact of introducing batch normalization is analyzed by examining the changes in accuracy and loss on the testing data.

Results analysis and interpretation are also critical parts of the approach. We must test the model's performance in

various settings to do this. The accuracy and loss rates of the model's digit prediction capabilities under different optimizers and learning rates are evaluated. You may learn much about the efficacy of multiple hyperparameters and training approaches from these comparisons. Its performance on specific test photos is further examined to assess the model's usefulness in real-world settings. A thorough evaluation of the results is the last step of the project. This involves analyzing the model's performance in different environments and determining the impact of different hyperparameters and architectural decisions. Understanding the role of various preprocessing stages, such as normalization and tensor conversion, in the model's performance is also explored in the analysis.

## IV. DATA ANALYSIS

Recognizing handwritten digits using a CNN applied to the MNIST dataset is the main emphasis of this research effort. The model was trained and tested using the MNIST dataset, a set of greyscale images of handwritten numbers. Each sample in the dataset is a 28x28 pixel image; there are 60,000 training examples and 10,000 test samples in total. Before CNN can do its job, the images undergo several changes, such as being normalized and converted to tensor format. The model uses training and testing data sets to sort the images into ten categories, each representing a number from zero to nine. Digit recognition is a fundamental problem in computer vision, and this project uses deep learning techniques with this classic dataset to tackle it.

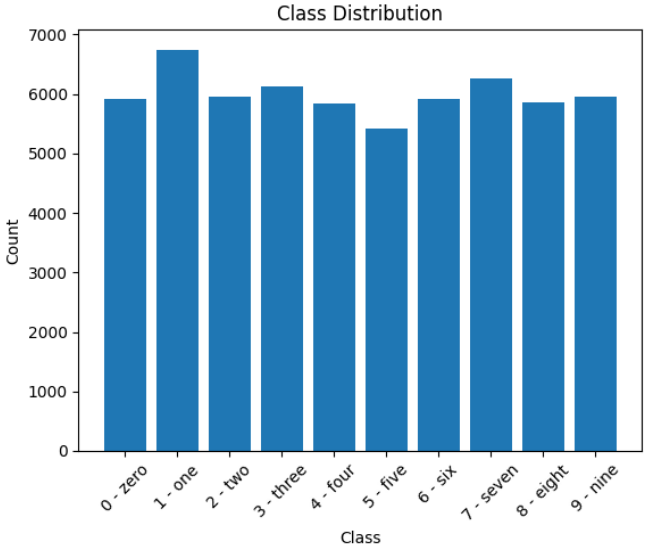


**Figure 1: Class Distribution**

Figure 1 shows the distribution of classes within that dataset. The dataset is reasonably balanced, with each category representing 0 to 9. The dataset has a nearly uniform representation of each digit class since the counts for each type are approximately equal and hover around the 6,000 level. The CNN model must train on an equivalent amount of data for each digit to improve the likelihood of reaching high accuracy across all classes during the recognition task. This uniformity facilitates this process.

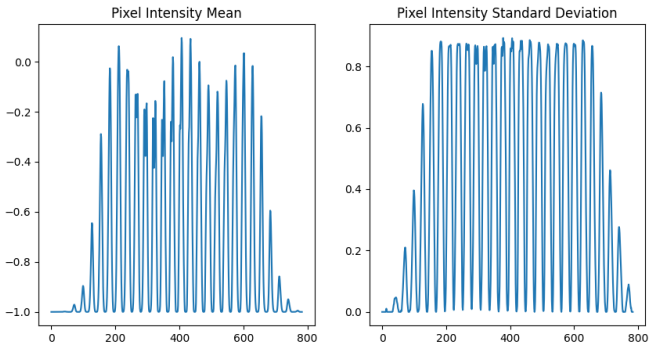


**Figure 2: Pixel Intensity Mean and Standard Deviation**

The two graphs in Figure 2 show pixel intensity statistics in the images from the MNIST dataset. The periodic pattern seen in the left chart, which shows the mean of pixel intensities, is related to the structure of the handwritten numbers, as darker parts (such as edges) are consistently associated with certain image sections. The right chart shows high variability at the same periodic sites, illustrating the pixel intensity standard deviation. The method to generate the Figure 2 graphs can be seen in the coding file.


**Figure 3: Training and Testing Data Class Distribution**

Figure 3 shows that each of the ten classes in the training data has many samples (5,000 to 6,000 per class), but there is a little inequity in the distribution of these samples (see left side of the figure). There is a more consistent distribution in the test data class distribution (shown on the right), with about 800 to 1,000 samples for each digit class.

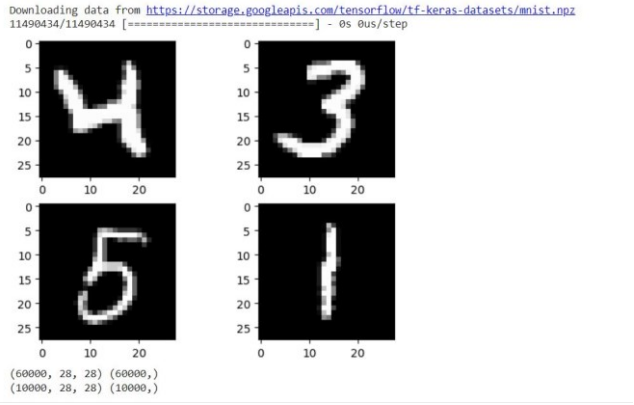


**Figure 4: Loading the Sample Images After Preprocessing**

Figure 4 shows that the images were converted to a four-dimensional array, and the pixel values were uniformly scaled to a range of [0, 1].

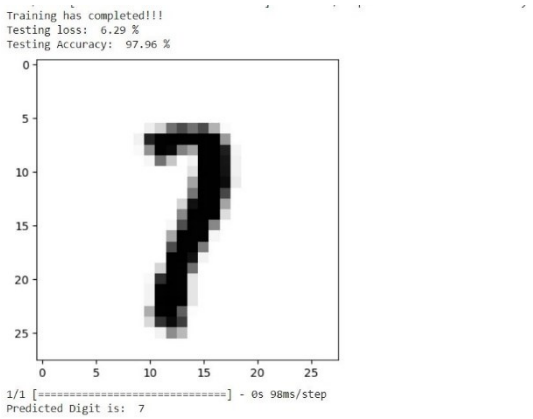


**Figure 5: Predicting the Digit with SGD Optimizer and 0.01 Learning Rate**

Figure 5 shows that we are testing the image. Based on the image, we know that the right answer is 7, and our trained model also predicts this, so this is fine. Also, the accuracy of our model is 97.96%, and the testing loss is only 6.29%, which is a good result.

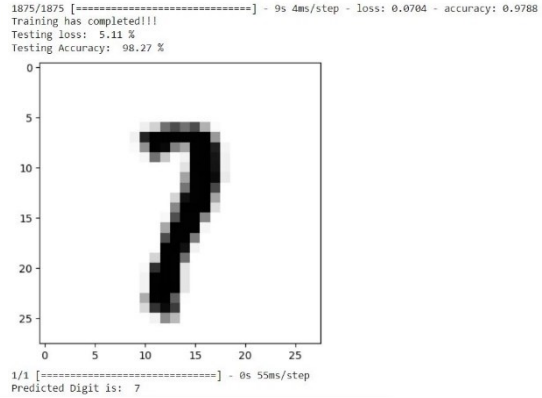


**Figure 6: Predicting the Digit with Adam Optimizer and 0.001 Learning Rate**

Figure 6 shows another version of 7, and it has been tested with different hyperparameters. It can be illustrated from the above figure that the testing accuracy is 98.27%, and the loss is only 5.51%, which is surely a better result than the previous result.

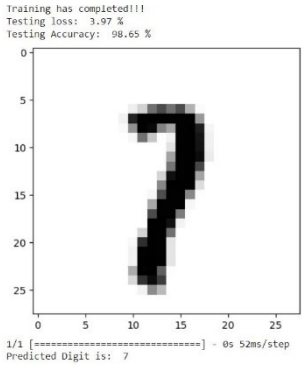


**Figure 7: Predicting the Digit with Adadelta Optimizer and 1.0 Learning Rate**

Just like above Figure 6, Figure 7 also shows that the hyperparameters have been changed, and just like above, the accuracy has improved, and the testing loss has decreased. The testing accuracy achieved is 98.65%, and testing loss has also been reduced to 3.97%.

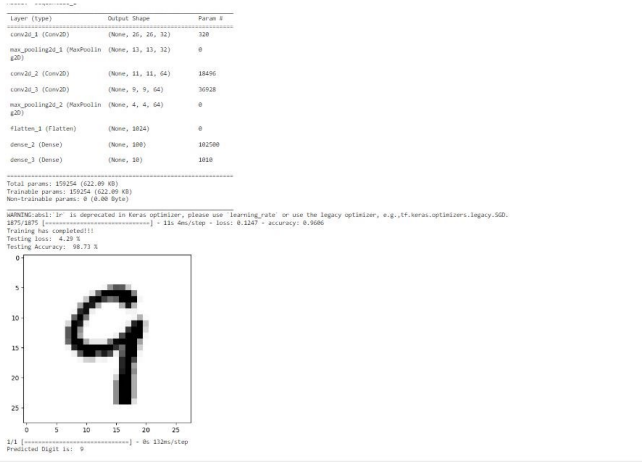


**Figure 8: Complex CNN**

A sophisticated CNN model is built and trained using the Keras Sequential model in Figure 8. This CNN model consists of two dense layers, a flattening step, and two convolutional and max pooling layers. The filters increase from 32 in the first convolutional layer to 64 in each succeeding layer. The model utilizes the 'he_uniform' kernel initializer and the 'ReLU' activation function. Using a learning rate of 0.01, SGD is the utilized optimizer. The model gets good results after being trained on the training dataset (X_train, Y_train). The model's efficacy in image classification is demonstrated by its testing loss and accuracy on the test dataset (X_test, Y_test). Furthermore, the model's ability to accurately estimate the digit in a particular test image (index 150) demonstrates its usefulness in image identification tasks. The model's intricacy and capacity to handle comprehensive image data are highlighted by its 159,254 trainable parameters, as revealed in the model overview. Also, for this complex CNN, the achieved testing accuracy is 98.73%, and the testing loss is 4.29%.


**Figure 9: Predicting the Digit with SGD Optimizer, 0.01 Learning Rate, and Batch Normalization**

Finally, in Figure 9, with batch normalization, the learning rate of 0.01, and SGD optimizer, the testing data accuracy was 99.24%, and the testing loss was only 2.63%.

## V. CONCLUSION

Results from this experiment show that Convolutional Neural Networks (CNNs) performed well on the MNIST dataset when it came to identifying handwritten numbers. The importance of careful preprocessing and optimizing the model architecture in attaining high accuracy in digit classification is demonstrated throughout the project lifecycle, from data preparation to model creation, training, and performance evaluation. The dataset was normalized, and the tensor was transformed to extract and understand the

complex patterns in handwritten numbers. Then, a CNN model was meticulously built using many convolutional, max pooling, and dense layers. Experiments using alternative learning rates of optimizers, such as SGD, Adam, and Adadelta, shed light on the learning dynamics of the model. Better accuracy and less loss are signs that the model's performance was already high before batch normalization was used. In the future, the use of deeper networks or CNNs with multiple layers can detect more complex patterns in the data that might be further investigated.

## REFERENCES


[1] S. Boopathi and U. K. Kanike, "Applications of Artificial Intelligent and Machine Learning Techniques in Image Processing," *IGI Global Publishing Tomorrow's Research Today,* 2023.

[2] S. İ. Omurca, E. Ekinci, S. Sevim, E. B. Edinç, S. Eken and A. Sayar, "A document image classification system fusing deep and machine learning models," *Applied Intelligence,* vol. 53, pp. 15295-15310, 2023.

[3] R. Thanki, "A deep neural network and machine learning approach for retinal fundus image classification," *Healthcare Analytics,* vol. 3, 2023.

[4] C. J. Haug and J. M. Drazen, "Artificial Intelligence and Machine Learning in Clinical Medicine," *The New England Journal of Medicine,* 2023.

[5] M. Rana and M. Bhushan, "Machine learning and deep learning approach for medical image analysis: diagnosis to detection," *Multimedia Tools and Applications,* vol. 82, p. 26731–26769, 2023.

[6] I. Cinar, Y. S. Taspinar, R. Butuner, R. Kursun, M. H. Calp and M. Koklu, "Classification of deep image features of lentil varieties with machine learning techniques," *European Food Research and Technology,* vol. 249, pp. 1303-1316, 2023.

[7] M. S. Rana , M. . H. Kabir and A. Sabour, "Comparison of the Error Rates of MNIST Datasets," *North American Academic Research,* vol. 6, no. 5, 2023.

[8] S. Gouraguine, . M. Qbadou, M. RAFIK and K. Mansouri, "A New Knowledge Primitive of Digits Recognition for NAO Robot Using MNIST Dataset and CNN Algorithm for Children's Visual Learning Enhancement," *Journal of Information Technology Education,* vol. 22, pp. 389-408, 2023.

[9] H. Shao, E. Ma, M. Zhu and X. Deng, "MNIST Handwritten Digit Classification Based on Convolutional Neural Network with Hyperparameter Optimization," *Intelligent Automation & Soft Computing,* vol. 36, no. 3, 2023.

[10] S. Pashine, R. Dixit and R. Kushwah, "Handwritten Digit Recognition using Machine and Deep Learning Algorithms," *Computer Vision and Pattern Recognition,* 2021.

[11] A. Baldominos, Y. Saez and P. Issai, "A Survey of Handwritten Character Recognition with MNIST and EMNIST," *Applied Sciences,* vol. 9, 2019.

[12] Y. Gong and P. Zhang, "Research on Mnist Handwritten Numbers Recognition based on CNN," *Journal of Physics: Conference Series,* vol. 2138, 2021.

[13] B. Gope, S. D. Pande, N. Karale and S. Dharmale, "Handwritten Digits Identification Using Mnist Database Via Machine Learning Models Handwritten Digits Identification Using Mnist Database Via Machine Learning Models," *IOP Conference Series Materials Science and Engineering,* vol. 1022, no. 1, 2021.

[14] Y. Peng, "Digital Recognition Methods Based on Deep Learning," *Scientific Programming,* 2022.

[15] S. Ali, Z. Shaukat, M. Azeem, Z. Sakhawa, T. Mahmood and K. u. Rehman, "An efcient and improved scheme for handwritten digit recognition based on convolutional neural network," *Springer Nature Switzerland AG,* 2019.

[16] R. Zhu, S. Lilak, A. Loeffler, J. Lizier, A. Stieg, J. Gimzewski and Z. Kuncic, "Online dynamical learning and sequence memory with neuromorphic nanowire networks," *Nature Communications,* 2023.

[17] P. H. Jain , V. Kumar, J. Samuel, S. Singh, A. Mannepalli and R. Anderson, "Artificially Intelligent Readers: An Adaptive Framework for Original Handwritten Numerical Digits Recognition with OCR Methods," *Information,* vol. 14, 2023.

[18] D. Jayswal, B. Y. Panchal, B. Patel and N. Acharya, "Study and Develop a Convolutional Neural Network for MNIST Handwritten Digit Classification," *Artificial Neural Networks,* pp. 407-416, 2022.

[19] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," *2017 International Conference on Engineering and Technology (ICET),* pp. 1-6, 2017.

[20] E. YAZAN and M. F. Talu, "Comparison of the stochastic gradient descent based optimization techniques," *International Artificial Intelligence and Data Processing Symposium (IDAP),* pp. 1-5, 2017.

[21] Z. Zhang, "Improved Adam Optimizer for Deep Neural Networks," *IEEE/ACM 26th International Symposium on Quality of Service (IWQoS),* pp. 1-2, 2018.

[22] M. S. Devi, R. Aruna, D. R. Rajeswari and R. S. Manogna, "Conv2D Xception Adadelta Gradient Descent Learning Rate Deep learning Optimizer for Plant Species Classification," *Third International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT),* pp. 1-4, 2023.

[23] R. Lin, "Analysis on the Selection of the Appropriate Batch Size in CNN Neural Network," *International Conference on Machine Learning and Knowledge Engineering (MLKE),* pp. 106-109, 2022.