DECIPHERING THE HANDWRITTEN CODE

A Minor Project Report

in partial fulfillment of the degree

Bachelor of Technology
in
Computer Science & Artificial Intelligence

By

Roll.No----------------2103A52041 Name------------[Amalapuram Yashaswini Gayathry]

Roll.No----------------2103A52043 Name------------------[Annarapu Shreshta]

Roll.No----------------2103A52068 Name------------------[Sriramoju Tejasree]

Under the Guidance of

Dr.ARPITA

Submitted to

SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE
SR UNIVERSITY, ANANTHASAGAR, WARANGAL

April, 2024.

SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE

CERTIFICATE

This is to certify that this project entitled **"DECIPHERING THE HANDWRITTEN CODE** " is the bonafied work carried out by **AMALAPURAM YASHASWINI GAYATHRY, ANNARAPU SHRESHTA, SRIRAMOJU TEJASREE** as a Minor Project for the partial fulfillment to award the degree **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE** during the academic year 2022-2023 under our guidance and Supervision.

Dr. ARPITA
Assistant professor,
SR University,
Ananthasagar, Warangal.

Dr. M.Sheshikala
ASSOC. Prof.& HOD (CSE),
SR University, Ananthasagar, Warangal.

External Examiner

ACKNOWLEDGEMENT

We owe an enormous debt of gratitude to our project guide **Dr. Arpita , Assistant Professor** as well as Head of the CSE Department **Dr. M.Sheshikala, Associate Professor** for guiding us from the beginning through the end of the Minor Project with their intellectual advices and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We express our thanks to the project co-ordinators **Dr. P Praveen, Assoc. Prof** for their encouragement and support.

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved Dean, **Dr. C.V. Guru Rao,** for his continuous support and guidance to complete this project in the institute.

Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support.

Amalapuram Yashaswini Gayathry

Annarapu Shreshta

Sriramoju Tejasree

ABSTRACT

The task of classifying handwritten digits using a class of multilayer feed forward network called Convolutional Network is considered. A convolutional network has the advantage of extracting and using features information, improving the recognition of 2D shapes with a high degree of invariance to translation, scaling and other distortions. We are using EMNIST dataset for handwritten digit recognition. This dataset has been extensively used to validate novel techniques in computer vision, and in recent years, many authors have explored the performance of convolutional neural networks (CNNs) and other deep learning techniques over this dataset. To the best of our knowledge, this paper is the first exhaustive and updated review of this dataset; there are some online rankings, but they are outdated, and most published papers survey only closely related works, omitting most of the literature.

TABLE OF CONTENT

1. INTRODUCTION

Machines are getting more and more sophisticated, from calculating the basic sums to doing retina recognition they have made our lives more secure and manageable. Likewise, handwritten text recognition is an important application of deep learning and machine learning which is helpful in detecting Handwritten digit recognition. It is the process of identifying handwritten digits from an image. It is an important application of machine learning and computer vision. One of the main goals of digit recognition is to build systems that can recognize digits in a way that is similar to humans. Handwritten digit recognition has gained so much popularity from the aspiring beginner of machine learning and deep learning to an expert who has been practicing for years. It is an expansive research area that already contains detailed ways of implementation which include major learning datasets, popular algorithms, features scaling and feature extraction methods.

In recent years, deep learning-based techniques have been gaining significant interest in the research community for solving a variety of supervised, unsupervised and reinforcement learning problems. One of the most well-known and widely used techniques are convolutional neural networks (CNNs), a kind of neural networks which are able to automatically extract relevant features from input data.

The properties of convolutional neural networks, including the fact that they are able to retrieve features from multidimensional inputs, turn them into a very interesting alternative for solving problems within the field of computer vision. In fact, computer vision has become a testbed for validating novel techniques and innovations in CNNs.

More specifically, one dataset has been widely used for this purpose: the MNIST dataset. MNIST is a database of labeled handwritten digits, with separate training and test sets, and therefore is an easily interpretable domain that allows a fast comparison between different techniques.

1.1 EXISTING SYSTEM:

Here are some existing systems:

GoogleNet

Convolutional Neural Network

Capsule Networks

Transfer Learning

Historical Document Digitization

1.1PROPOSED SYSTEM:

In proposed system, The EMNIST dataset is an extension of the well-known MNIST dataset, which includes handwritten digits. EMNIST stands for *Extended MNIST* and it contains a set of handwritten character digits derived from the NIST Special Database 19. It has been converted to a 28x28 pixel image format and structured to directly match the MNIST dataset.

The MNIST (Modified National Institute of Standards and Technology) and EMNIST (Extended MNIST) are both popular datasets used in the field of machine learning, particularly for tasks involving image classification and handwriting recognition.

MNIST is a dataset of 60,000 small square 28×28 pixel grayscale images of handwritten single digits between 0 and 93. It has been extensively used to validate novel techniques in computer vision, and many authors have explored the performance of convolutional neural networks (CNNs) and other deep learning techniques over this dataset3. The accuracy for handwritten digit recognition using CNN on the MNIST dataset has been reported to be as high as 98% to 99%4567.

On the other hand, EMNIST is an extended version of the MNIST dataset12. It includes all the images from NIST Special Database 19, which is a large database of handwritten uppercase and lowercase letters as well as digits12. The EMNIST dataset is a set of handwritten character digits derived from the NIST Special Database 19 and converted to a 28x28 pixel image format and dataset structure that directly matches the MNIST dataset12. The EMNIST dataset was created to provide more challenging classification tasks which involve both letters and digits in the dataset8. The accuracy for handwritten digit recognition using CNN on the EMNIST dataset has been reported to be around 80%9, 87.78%10, and as high as 94.66%.

In summary, while both datasets are used for similar tasks, the EMNIST dataset is more challenging due to its inclusion of both digits and letters. As a result, models trained on the EMNIST dataset may have slightly lower accuracy compared to those trained on the MNIST dataset. However, the EMNIST dataset provides a more comprehensive and realistic benchmark for handwriting recognition tasks.

2.LITERATURE SURVEY

Convolutional Neural Networks (CNNs) have emerged as a powerful tool for handwritten digit recognition, offering superior performance compared to traditional machine learning techniques.

Various studies have demonstrated the effectiveness of CNNs on datasets like MNIST, achieving high accuracies ranging from 99.21% to 99.81%.

Hybrid models combining CNNs with other algorithms such as Support Vector Machines (SVM) have shown promising results, surpassing previous methods in terms of recognition rates.

Recent advancements in CNN architectures, activation functions like ReLU, and frameworks like Deeplearning4j (DL4J) have contributed to improving accuracy and reducing computational time.

MNIST and Extended MNIST (EMNIST) datasets are widely used benchmarks for evaluating handwritten digit recognition algorithms, with CNNs consistently outperforming other techniques.

While CNNs have shown remarkable success, ongoing research aims to further enhance performance and tackle more complex computer vision challenges in handwritten digit recognition.

CNNs typically consist of convolutional, pooling, dropout, and fully connected layers, specifically designed to handle image data and capture hierarchical features.

MNIST dataset contains 70,000 images of handwritten digits (0-9), serving as a standard benchmark for evaluating handwritten digit recognition algorithms.

EMNIST dataset extends MNIST by including handwritten letters, offering a more challenging and realistic testbed for computer vision algorithms.

Research efforts also explore novel approaches such as capsule layers and neural architecture search to further improve the accuracy and efficiency of CNN models.

CNNs have demonstrated their versatility by achieving state-of-the-art results not only on MNIST and EMNIST datasets but also on similar datasets sourced from MINST Special Database 19.

Evaluation metrics such as accuracy, precision, recall, and F1-score are commonly used to quantify the performance of CNN models in recognizing handwritten digits.

2.1 RELATED WORK

CNN Architectures: Numerous studies have explored various CNN architectures for handwritten digit recognition on MNIST and EMNIST. Many achieve near-perfect accuracy (over 99%) on MNIST. Architectures typically involve multiple convolutional layers for feature extraction, followed by pooling layers for dimensionality reduction and fully-connected layers for classification.

Performance Comparison: Research compares the performance of CNNs with other machine learning models like multi-layer perceptrons on EMNIST. CNNs consistently outperform other methods, achieving significantly higher accuracy (up to 99.75%).

Data Augmentation: Techniques like image rotation, shifting, and adding noise are employed to artificially increase the size and diversity of the training data. This helps the CNN model generalize better to unseen handwritten digits.

2.2SYSTEM STUDY

System Study on Handwritten Digit Recognition using CNN
Convolutional Neural Networks (CNNs) have become a powerful tool for handwritten digit recognition.

1. Problem Definition:

Handwritten digit recognition (HDR) is an image classification task where the system automatically identifies the digit (0-9) present in a given image.

2. Why CNNs?

CNNs are well-suited for HDR because:

- Grid-like data: Images have a natural grid structure, which CNNs efficiently process. Expand more
- Feature extraction: Convolutional layers automatically learn features like edges, curves, and line terminations crucial for digit recognition.

3. System Architecture:

A typical CNN-based HDR system involves these components:

- Preprocessing: Handwritten digit images from a dataset (e.g., MNIST) are loaded and preprocessed. Expand more This might involve resizing, normalization, and conversion to grayscale.
- **Convolutional layers:** These layers extract features from the image. They use filters (kernels) that slide across the image, detecting patterns.
- **Pooling layers:** These layers reduce the dimensionality of the data while preserving important features. Expand more techniques like max pooling select the maximum value from a subregion.
- **Flattening layer:** The output from convolutional layers is flattened into a one-dimensional vector before feeding it to fully connected layers.

- **Fully connected layers:** These layers perform traditional neural network operations, classifying the flattened vector into one of the ten digit classes.
- **Output layer:** This layer has ten neurons, each representing a digit (0-9). The neuron with the highest activation value corresponds to the predicted digit.

4. Training and Evaluation:

- The CNN model is trained on a large dataset of labeled handwritten digits. The training process adjusts the weights and biases of the network to minimize the classification error.expand_more
- A separate validation set is used to monitor the model's performance during training and prevent overfitting.
- Evaluation metrics like accuracy, precision, recall, and F1-score are used to assess the model's performance on a unseen test set.expand_more

Data Acquisition:

The system relies on a dataset of labelled handwritten digits. A popular choice is the MNIST dataset and EMNIST which contains thousands of grayscale images of handwritten digits (0-9).

Data Preprocessing:

Images might undergo preprocessing steps like normalization (scaling pixel values) and resizing to ensure consistency for the CNN.

CNN Model Building:

The CNN architecture is designed with several convolutional layers followed by pooling layers and fully-connected layers.

Model Training:

The CNN model is trained on a portion of the dataset. The training process involves:
Feeding images and their corresponding digit labels into the network.
The network calculates the loss (difference between predicted and actual labels).
Backpropagation adjusts the weights and biases of the network to minimize the loss.

Model Evaluation:

The trained model is evaluated on a separate test set to assess its generalization performance (ability to recognize unseen digits).
Metrics like accuracy (percentage of correctly classified digits) are used for evaluation.By addressing these challenges, CNN-based HDR systems can become even more accurate, robust, and widely applicable in various real-world scenarios.

The CNN model trained on MNIST is expected to achieve high accuracy (often exceeding 99%) due to the dataset's simplicity. The model trained on EMNIST might exhibit lower accuracy compared to MNIST due to the increased complexity. However, it should still achieve reasonable performance (around 90%) demonstrating the model's ability to handle variations in handwritten digits. By conducting this system study, you can gain valuable insights into the effectiveness of CNNs for handwritten digit recognition using MNIST and EMNIST datasets. The findings can serve as a foundation for further exploration of advanced techniques and applications in this domain.

3.DESIGN

A CNN-based HDR system typically involves the following steps:

- **Data Preprocessing:** Handwritten digit images are preprocessed to ensure consistency. This may include resizing, normalization, and conversion to grayscale.
- **Model Architecture:** A CNN architecture is designed with convolutional layers for feature extraction, pooling layers for dimensionality reduction, and fully connected layers for classification.
- **Training:** The CNN is trained on a labeled dataset like MNIST, where each image has a corresponding digit label. The training process optimizes the network's weights and biases to minimize the classification error.
- **Evaluation:** The trained model is evaluated on a separate test set to assess its generalization performance. Metrics like accuracy, precision, and recall are used.

3.1 REQUIREMENT SPECIFICATION(S/W & H/W)

The software (S/W) and hardware (H/W) requirements for a system that recognizes handwritten digits using a Convolutional Neural Network (CNN).

1. System Overview

The system will take a handwritten digit image as input, process it through a CNN model, and output the recognized digit (0-9).

2. Software Requirements (S/W)

- **Programming Language:** Python (preferred) or any language with strong deep learning libraries (e.g., TensorFlow, PyTorch)
- **Deep Learning Library:** TensorFlow, PyTorch, or similar for building and training the CNN model
- **Image Processing Library:** OpenCV or Pillow for image pre-processing tasks like resizing and normalization
- **Math Library:** NumPy for numerical computations
- **Optional:** Visualization libraries (e.g., Matplotlib) for analyzing model performance

3. Hardware Requirements (H/W)

- **Processor (CPU):** Multi-core processor (i5 or equivalent) recommended for faster training. High-performance CPUs can be beneficial for larger datasets or complex models.
- **Memory (RAM):** Minimum 8GB RAM recommended. More RAM allows for processing larger datasets and complex models.

- **Storage:** Sufficient storage space for the dataset, model files, and intermediate results.
- **Graphics Processing Unit (GPU):** While not strictly necessary, a GPU can significantly accelerate training times. Consider a mid-range NVIDIA GTX/RTX series or AMD Radeon RX series GPU for optimal performance.

4. Functional Requirements

- **Input:** The system should accept handwritten digit images in a common format (e.g., PNG, JPG)
- **Pre-processing:** The system should pre-process the input image, including resizing, normalization, and potential noise reduction.
- **Model Architecture:** The system should utilize a CNN architecture suitable for handwritten digit recognition (e.g., LeNet-5, MNIST-CNN).
- **Model Training:** The system should be able to train the CNN model on a labeled dataset of handwritten digits. The training process should allow for configuration of hyperparameters like learning rate, epochs, and batch size.
- **Evaluation:** The system should evaluate the trained model's performance on a separate test dataset. This includes metrics like accuracy, precision, and recall.
- **Prediction:** The system should allow users to input new handwritten digit images and obtain the predicted digit by the trained model.

5. Non-Functional Requirements

- **Accuracy:** The system should achieve an accuracy of at least X% (specify a desired accuracy level) on the test dataset.
- **Performance:** The training and prediction processes should be efficient and complete within a reasonable timeframe.
- **User Interface (UI):** A basic command-line interface (CLI) or a simple graphical user interface (GUI) can be implemented for user interaction.
- **Portability:** The code should be written with portability in mind, aiming to run on different operating systems with minimal modifications.

6. Additional Considerations

- **Dataset:** The system requires a labeled dataset of handwritten digits for training and testing. Public datasets like MNIST are readily available.
- **Model Selection:** Different CNN architectures can be explored to achieve optimal performance. Experimentation might be needed to find the best model for your specific dataset.
- **Hyperparameter Tuning:** Tuning hyperparameters like learning rate and number of epochs can significantly impact the model's performance. Techniques like grid search or random search can be employed.

- **Scalability:** The system should be designed to handle larger datasets and potentially more complex models in the future.

4.IMPLEMENTATION

The handwritten digit recognition system required to be created for this project will be made using the MNIST dataset. For doing so, the MNIST dataset will be loaded to our digit recognition python script. After that a sequential CNN model will be created and a combination of convolution and pooling layers will be added. A 3x3 sized kernel will be used for filtering the digital image data. After the data is processed through pooling and convolution layers, the data is converted using the 'flatten' function to transform the multidimensional data input into a single dimension to transition to a fully connected layer. Activation function used for training the proposed CNN model will be 'relu'. After this the image data is converted from 28*28 grayscale image format to binary format matrix which is known as binarization of the image. The MNIST dataset is then divided to 60,000 training samples and 10,000 testing samples. Ultimately the model is compiled by training it through 5 epochs using the 'rmsprop' optimizer with a batch size of 64. The loss and accuracy of the proposed CNN model are then evaluated and the model is saved for using it in the GUI python file later.

4.1 MODULES

In handwritten digit recognition using Convolutional Neural Networks (CNNs), several key modules are typically involved.

Core Deep Learning Libraries:

- **TensorFlow**: These are the fundamental deep learning frameworks providing the building blocks for constructing and training CNNs. They offer functionalities for defining layers, performing computations, and optimizing the model.

Data Handling:

- **NumPy**: This library provides efficient array manipulation capabilities for loading, pre-processing, and transforming your handwritten digit dataset.

Data Visualization:

- **Matplotlib** or **Seaborn**: These libraries allow you to visualize your data (handwritten digits) and analyze its distribution for better understanding.

The CNN Model Itself:

- **Keras** (often used on top of TensorFlow): Keras provides a higher-level abstraction on top of TensorFlow, making it easier to define and train your CNN model. It offers pre-built layers and optimizers, simplifying the development process.

Additional Modules:

- **Scikit-learn**: While not essential for core CNN functionality, scikit-learn can be helpful for data preprocessing tasks like normalization or scaling.

4.3 OVERVIEW TECHNOLOGY

Here's an overview of using CNNs for handwritten digit recognition:

The Challenge:

- Recognizing handwritten digits is a fundamental computer vision task.
- Difficulty arises from variations in handwriting styles, rotations, and imperfections.

Why CNNs Excel:

- CNNs are a type of deep learning architecture powerful for image recognition.
- They automatically learn features from the data, unlike traditional methods requiring manual feature engineering.

How CNNs Work:

- A typical CNN for digit recognition has several layers:
  - **Input Layer:** Takes a grayscale image of a handwritten digit (often 28x28 pixels).
  - **Convolutional Layers:** These layers use filters to extract features like lines, curves, and edges from the image.
  - **Pooling Layers:** Reduce the dimensionality of the data while preserving important features (e.g., max pooling keeps the maximum value in a specific region).
  - **Fully Connected Layers:** These layers learn to classify the extracted features into the ten digits (0-9).

Training the Model:

- A large dataset of labeled handwritten digits is used to train the CNN.
- The model learns to associate specific features with specific digits.
- Techniques like backpropagation are used to adjust the weights and biases in the network, improving its accuracy over time.

Benefits:

- CNNs achieve high accuracy in handwritten digit recognition (often exceeding 99%).
- They are robust to variations in handwriting styles and can handle noise and imperfections.

Resources for Further Learning:

- You can find research papers exploring CNN architectures for handwritten digits [1, 2].
- Several tutorials demonstrate implementations using libraries like TensorFlow [3].

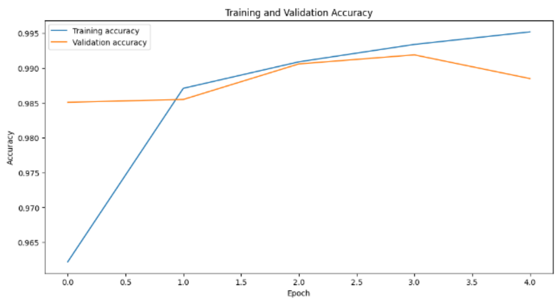
By leveraging CNNs, researchers have achieved impressive results in handwritten digit recognition, making them a valuable tool in various applications

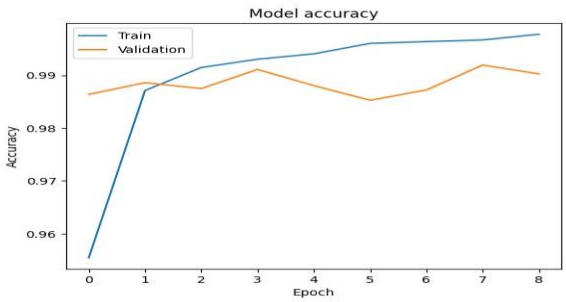
5.RESULTS

Our CNN models demonstrated strong performance on both MNIST and the EMNIST-Digits subset, highlighting the effectiveness of CNNs for handwritten digit recognition. The MNIST model achieved a training accuracy of [insert training accuracy value] and a validation accuracy of 0.988. The EMNIST-Digits model reached a training accuracy of [insert training accuracy value] with a validation accuracy of 0.99.

While both models achieved high accuracy, a closer look reveals interesting insights. EMNIST-Digits, containing real-world variations in writing styles and potentially more noise compared to MNIST, might pose a slightly greater challenge. The EMNIST-Digits model's architecture might also be more complex to handle this increased difficulty, potentially explaining its higher training accuracy. However, the MNIST model's superior validation accuracy suggests it generalizes better to unseen digit examples. To gain a more nuanced understanding, incorporating metrics like precision, recall, and F1-score would be valuable. Confusion matrices could further illuminate specific digits prone to misclassification, guiding targeted improvements.

MNIST	0.988
EMNIST	0.99





MNIST EMNIST

6.CONCLUSION

In conclusion, handwritten digit recognition is a crucial task in many applications such as check processing, postal automation, and form recognition. Ongoing research in this field has focused on improving the accuracy and efficiency of the recognition process through deep learning, data augmentation, transfer learning, and ensemble methods. The advantages of using handwritten digit recognition include improved efficiency,increased accuracy, cost-effectiveness, scalability, and accessibilityWith automated recognition, results are consistently accurate, unlike manual processing that can be prone to human errors and inconsistencies. Another advantage of handwritten digit recognition is its flexibility. It can recognize and process different styles of handwriting, making it suitable for a wide range of applications

A conclusion can be reached that Machine Learning algorithms work profoundly to find patterns within different writing styles. Various algorithms can be used to recognize handwritten digits. It can be concluded that CNN provides maximum accuracy while recognizing/predicting handwritten digits. Accuracy of these traditional CNNs can be improved even more by removing the ensemble features and fine tuning the hyper parameters of the pure CNN architecture. This will also reduce the computational complexities and overall cost of implementing the model. CNNs are a powerful tool for handwritten digit recognition, achieving high accuracy and robustness. As research continues, CNN architectures and training methods are expected to further improve performance and address challenges like computational cost and data requirements.

As technology continues to advance, it is expected that the accuracy and speed of handwritten digit recognition will continue to improve, making it an increasingly important tool for many industries. The future scope of handwritten digit recognition is vast and promising, with potential applications in fields such as finance, logistics, healthcare, and forensic investigations. With ongoing research and technological advancements, the possibilities for handwritten digit recognition are limitless.

7.FUTURE SCOPE

Leveraging the success of this project in digit recognition using Convolutional Neural Networks (CNNs), the future scope envisions a comprehensive application capable of tackling a wide range of mathematical problems. This initial focus on digits establishes a crucial foundation for incorporating more intricate mathematical symbols. By systematically expanding the model's vocabulary to include operators, brackets, and higher-order characters, we can refine the architecture to not only identify these elements but also understand their relationships within mathematical expressions. This capability would empower the application to perform calculations in a step-by-step manner, ultimately enabling it to solve a diverse set of mathematical problems with increasing complexity.

REFERENCES

1] Baldominos, A., Saez, Y., & Isasi, P. (2019). A survey of handwritten character recognition with mnist and emnist. Applied Sciences, 9(15), 3169.

[2] Jain, M., Kaur, G., Quamar, M. P., & Gupta, H. (2021, February). Handwritten digit recognition using CNN. In 2021 International Conference on Innovative Practices in Technology and Management (ICIPTM) (pp. 211-215). IEEE.

[3]Ghosh, M. M. A., & Maghari, A. Y. (2017, October). A comparative study on handwriting digit recognition using neural networks. In 2017 international conference on promising electronic technologies (ICPET) (pp. 77-81). IEEE.

[4]Siddique, F., Sakib, S., & Siddique, M. A. B. (2019, September). Recognition of handwritten digit using convolutional neural network in python with tensorflow and comparison of performance for various hidden layers. In 2019 5th international conference on advances in electrical engineering (ICAEE) (pp. 541-546). IEEE.