PREPARED BY
ZIAN Hafsa
EDDRIOUCH Aya

PREPARED FOR
ELYAZIDI Youness

Handwritten Digit Recognition using Neural Networks

2024/2025

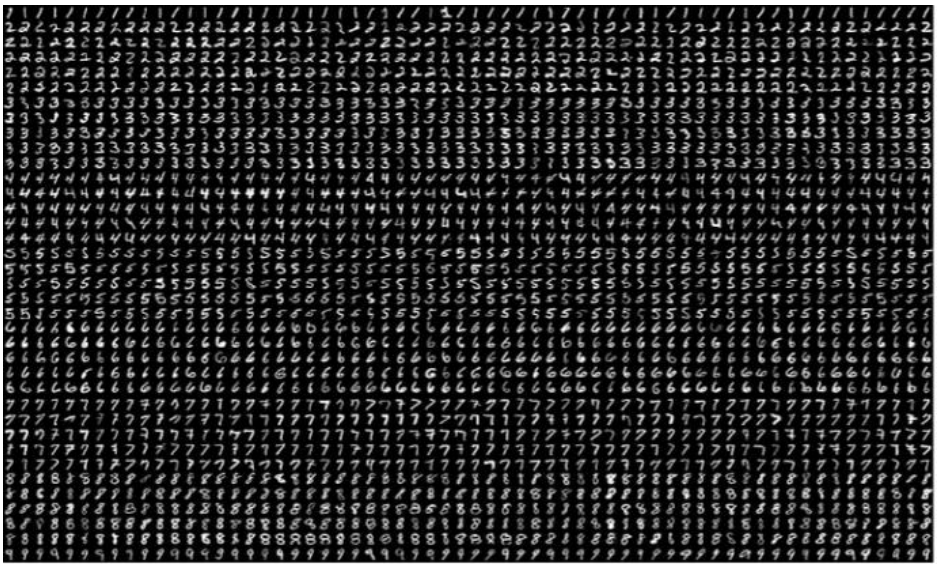
Contents

1.Introduction

Handwritten digit classification is a fundamental problem in the field of computer vision and machine learning, with applications in areas such as postal code recognition, banking, and document analysis. Initially, we attempted to use a simple neural network for this task, but it failed to provide accurate predictions due to its inability to capture spatial features. This led us to adopt Convolutional Neural Networks (CNNs), which are specifically designed to handle image data.

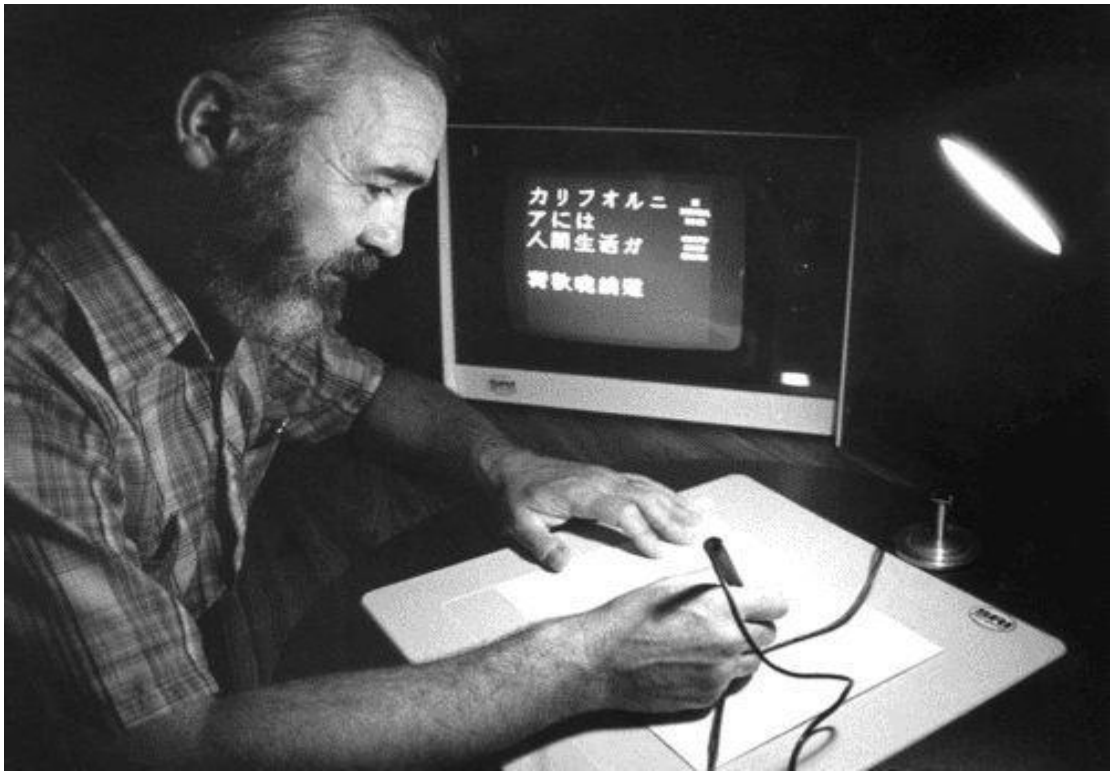
In the convolutional wireless data transmission process, and to overcome the challenges of data transmission and reception, a newly developed technology has been utilized. In the field of image processing, digitization has played a major role in processing data, information, and digital files at a lower cost compared to traditional paper processing. The benefit of digitization includes the development of handwriting recognition systems that convert handwritten characters into a readable digital format.

This technology has applications in fields such as archaeology, where it helps preserve old documents in libraries and banks, and education, with tools like digital dictionaries and handwriting recognition systems. Additionally, handwriting recognition has found applications in robotics. These applications deal with large amounts of data, requiring high accuracy in identification. Convolutional Neural Networks have proven to be highly effective, achieving accurate results with low error rates, distinguishing them from earlier methods used for handwritten digit recognition.



2. Historical Development of Handwritten Digit Recognition

The field of handwritten digit recognition has a rich history rooted in the advancement of pattern recognition and machine learning. Early methods relied on handcrafted features and statistical techniques. For instance, techniques such as template matching and feature extraction (e.g., edge detection, pixel intensity histograms) were common in the mid-20th century. These methods, while effective for simple tasks, struggled with large datasets or variations in handwriting styles.



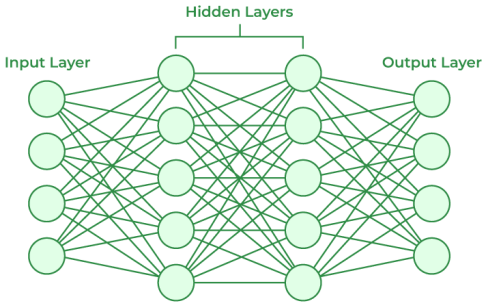
The introduction of artificial neural networks (ANNs) in the late 20th century revolutionized this field. In 1989, Yann LeCun and his team developed the LeNet-5 architecture, one of the earliest Convolutional Neural Networks (CNNs). It demonstrated remarkable success in recognizing handwritten digits, especially on the MNIST dataset, which became the standard benchmark for this task.

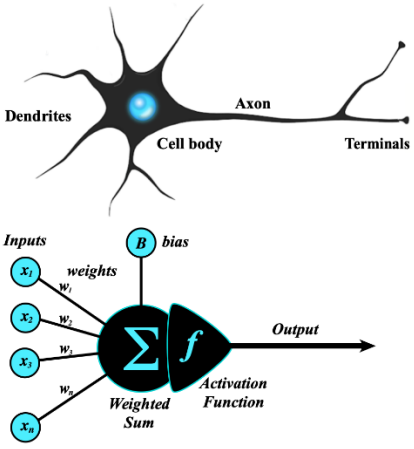
With the advent of more powerful hardware (e.g., GPUs) and the availability of large datasets, deep learning techniques such as CNNs have become the state-of-the-art approach. These models surpass traditional methods by automatically learning hierarchical features from data, eliminating the need for manual feature engineering. Today, handwritten digit recognition systems are highly accurate, with applications in banking, postal services, and beyond.

3. Neural Networks

Neural networks are a class of machine learning algorithms inspired by the structure and functioning of the human brain. They consist of interconnected layers of artificial neurons; each designed to process and transform data. A typical neural network includes an input layer, one or more hidden layers, and an output layer. The network learns by adjusting the weights and biases associated with each connection using a process called backpropagation.

While simple neural networks, often referred to as fully connected or dense networks, are effective for structured data, they struggle with image data due to their inability to capture spatial relationships. These limitations have led to the development of specialized architectures, such as Convolutional Neural Networks (CNNs), which are tailored for image-based tasks. Neural networks have become a cornerstone of modern AI, enabling advancements in fields like computer vision, natural language processing, and robotics.





4. Handwritten Digit Datasets

4.1. MNIST Dataset

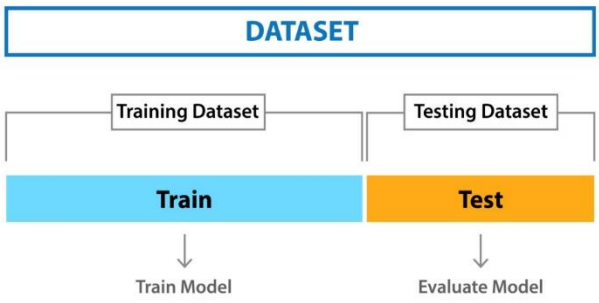
The MNIST dataset is a collection of 70,000 grayscale images of handwritten digits, each of size 28x28 pixels. It is divided into:

- **Training set:**

  60,000 images

- **Test set:**

  10,000 images

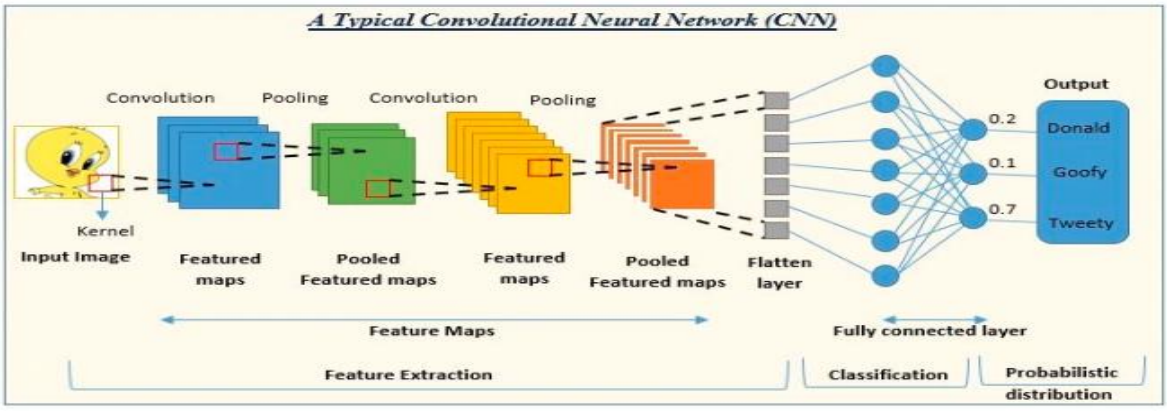


Each image is labeled with its corresponding digit (0-9). Before training the model, the pixel values were normalized to a range of 0 to 1 to ensure faster and more efficient learning.

4.2. Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a type of neural network designed specifically for image-based tasks. They use **convolutional layers** to detect features like edges and textures by applying filters across the image. **Pooling layers** reduce the spatial dimensions, making the model efficient and robust to small variations. After extracting features, the data is flattened and passed to **fully connected layers** for classification.

CNNs are highly efficient due to weight sharing, which reduces the number of parameters, making them less prone to overfitting. Their ability to automatically learn features has revolutionized computer vision, enabling applications in facial recognition, autonomous vehicles, and more.



5. Preprocessing Techniques

To ensure the efficiency and accuracy of our Convolutional Neural Network (CNN) model for handwritten digit recognition, several preprocessing steps were applied to the MNIST dataset:

5.1. Normalization of Pixel Values

The raw pixel values of the images range from 0 to 255. These values were normalized by dividing them by 255 to scale them into the range [0, 1]. This step ensures consistent input values, improves numerical stability, and speeds up the training process.

5.2. Reshaping the Images

The MNIST dataset contains 28x28 grayscale images. To adapt the data to the CNN, an additional channel dimension was added, resulting in a shape of 28x28x1. This allows the CNN to properly interpret the images as single-channel inputs.

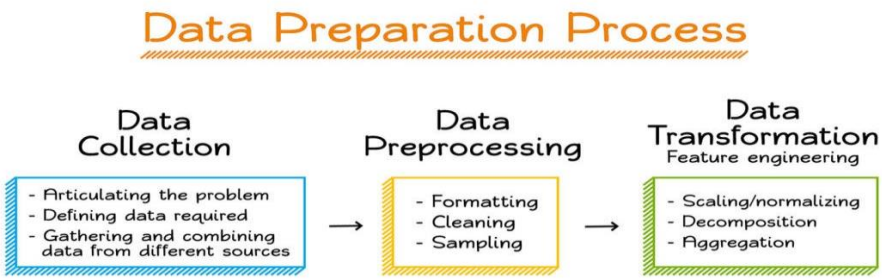
5.3. One-Hot Encoding of Labels

The labels (digits from 0 to 9) were converted into one-hot encoded vectors. For instance, the label "3" becomes [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]. This format is necessary for multi-class classification tasks using a softmax output layer.

5.4. Data Splitting

The dataset was divided into two parts: 60,000 images for training and 10,000 images for testing. This split allows us to train the model on a large set of data while reserving unseen data for evaluating its generalization performance.

⇒ These preprocessing steps were critical in ensuring that the CNN could effectively learn meaningful features from the data and achieve high accuracy on the handwritten digit recognition task.



6. Model Architecture

The architecture of the CNN used in this project consists of the following layers:

1. **Convolutional Layer 1**:

   32 filters with a kernel size of (3, 3), ReLU activation, and input shape (28, 28, 1).

2. **Max Pooling Layer 1**:

   Pool size (2, 2) to reduce spatial dimensions.

3. **Convolutional Layer 2**:

   64 filters with a kernel size of (3, 3), ReLU activation.

4. **Max Pooling Layer 2**:

   Pool size (2, 2).

5. **Flatten Layer**:

   Converts the 2D feature maps into a 1D vector.

6. **Fully Connected Layer**:

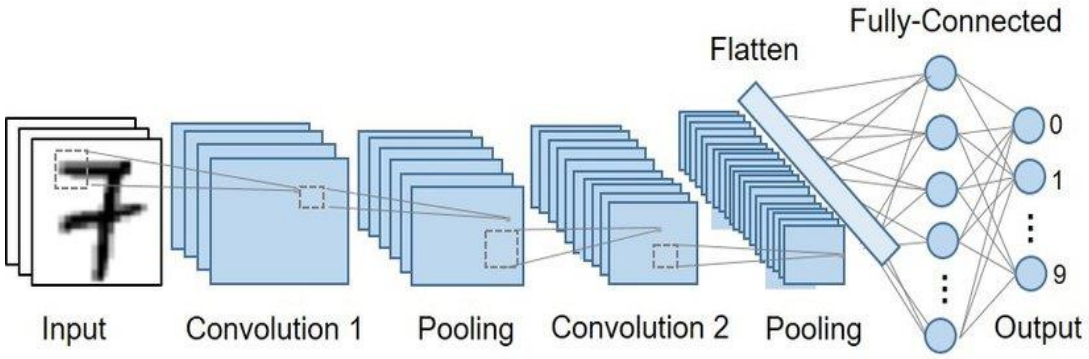
   64 neurons, ReLU activation, and a Dropout rate of 0.5 for regularization.

7. **Output Layer**:

   10 neurons (one for each digit), softmax activation for classification.

The model was compiled using the Adam optimizer, categorical crossentropy loss, and accuracy as the

evaluation metric.



7.Training and Evaluation

The model was trained for 10 epochs with a batch size of 64. The training and validation accuracies were monitored to ensure the model generalized well to unseen data. After training, the model achieved:

- **Test Accuracy**: **~98.7% & ~99%**

This indicates the model's robustness and its ability to correctly classify digits.

8.Conclusion and Summary

The model was evaluated on randomly selected test images, and the predictions were visualized alongside their true labels to assess performance. The results showed that most predictions were accurate, with only a few misclassifications. This highlights the reliability of the CNN for handwritten digit recognition.

Additionally, the training and validation accuracy consistently improved over the epochs, reflecting the model's capacity to learn meaningful features from the dataset. Similarly, the training and validation loss decreased steadily, indicating effective optimization and minimal overfitting.

This project successfully demonstrated the application of Convolutional Neural Networks (CNNs) for handwritten digit classification, achieving high accuracy and generalization capabilities. These results confirm the strength of CNNs in tackling image classification tasks and their potential for real-world applications, such as automated data entry, digital archiving, and educational tools.

Future improvements could involve exploring advanced techniques such as data augmentation to increase the diversity of the training data, implementing deeper network architectures to capture more complex features, or leveraging transfer learning to further enhance model performance. These steps could make the system even more robust and adaptable to a wider range of handwritten digit recognition challenges.

9.References

[1] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

[2] MNIST Dataset: http://yann.lecun.com/exdb/mnist/

[3] PyTorch Tutorials: **Training a Classifier**
https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html

[4] TensorFlow Documentation: **Building Convolutional Neural Networks**
https://www.tensorflow.org/tutorials/images/cnn?hl=fr

[5] Towards Data Science: **A Beginner's Guide to CNNs**
https://towardsdatascience.com/convolution-neural-networks-a-beginners-guide-implementing-a-mnist-hand-written-digit-8aa60330d022

[6] Goodfellow, I., Bengio, Y., & Courville, A. (2016). **Deep Learning**. MIT Press.

[7] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). **ImageNet classification with deep convolutional neural networks**. Advances in Neural Information Processing Systems (NIPS), 1097-1105.

10. Appendices

Technical Configuration

Details of the tools and libraries used to complete the project:

- **Programming Language**: Python 3.10
- **Framework**: TensorFlow 2.12.0
- **Key Libraries**: NumPy, Matplotlib
- **Hardware**:
  - **Processor**: Intel i5-1035G1
  - **RAM**: 8 GB
  - **GPU**: Not used (training performed on CPU)