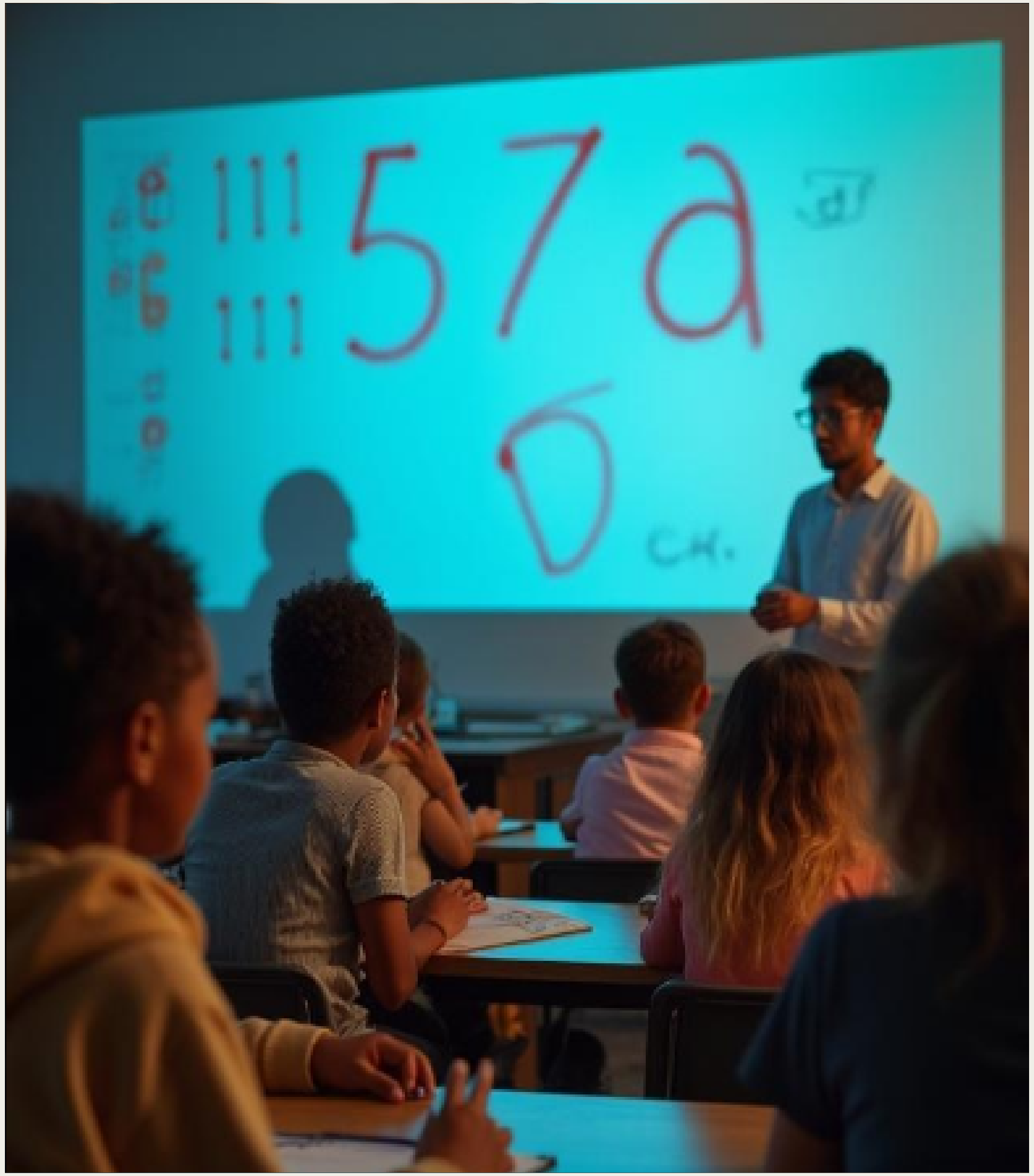
Handwritten digit recognition(MNIST)

Building and Evaluating a Convolutional Neural Network on the MNIST Dataset

22R11A6770
22R11A6756
23R11A62J6

Introduction

The handwritten digit recognition project employs Artificial Intelligence and Machine Learning (AIML) to create a robust system for identifying and classifying handwritten digits (0-9) from digital images. By utilizing the MNIST dataset, a standard benchmark containing 28x28 grayscale images of handwritten digits, the project implements advanced techniques such as Convolutional Neural Networks (CNNs) or other supervised learning algorithms. The process involves data preprocessing (e.g., normalization, noise reduction), feature extraction, model training, and performance evaluation to achieve high accuracy across diverse handwriting styles. The system aims to address challenges like variability in writing patterns, noise, and image distortions. Applications include automated postal code recognition, bank check processing, and digit-based authentication systems. The project emphasizes model optimization, hyperparameter tuning, and validation to ensure scalability and real-world applicability



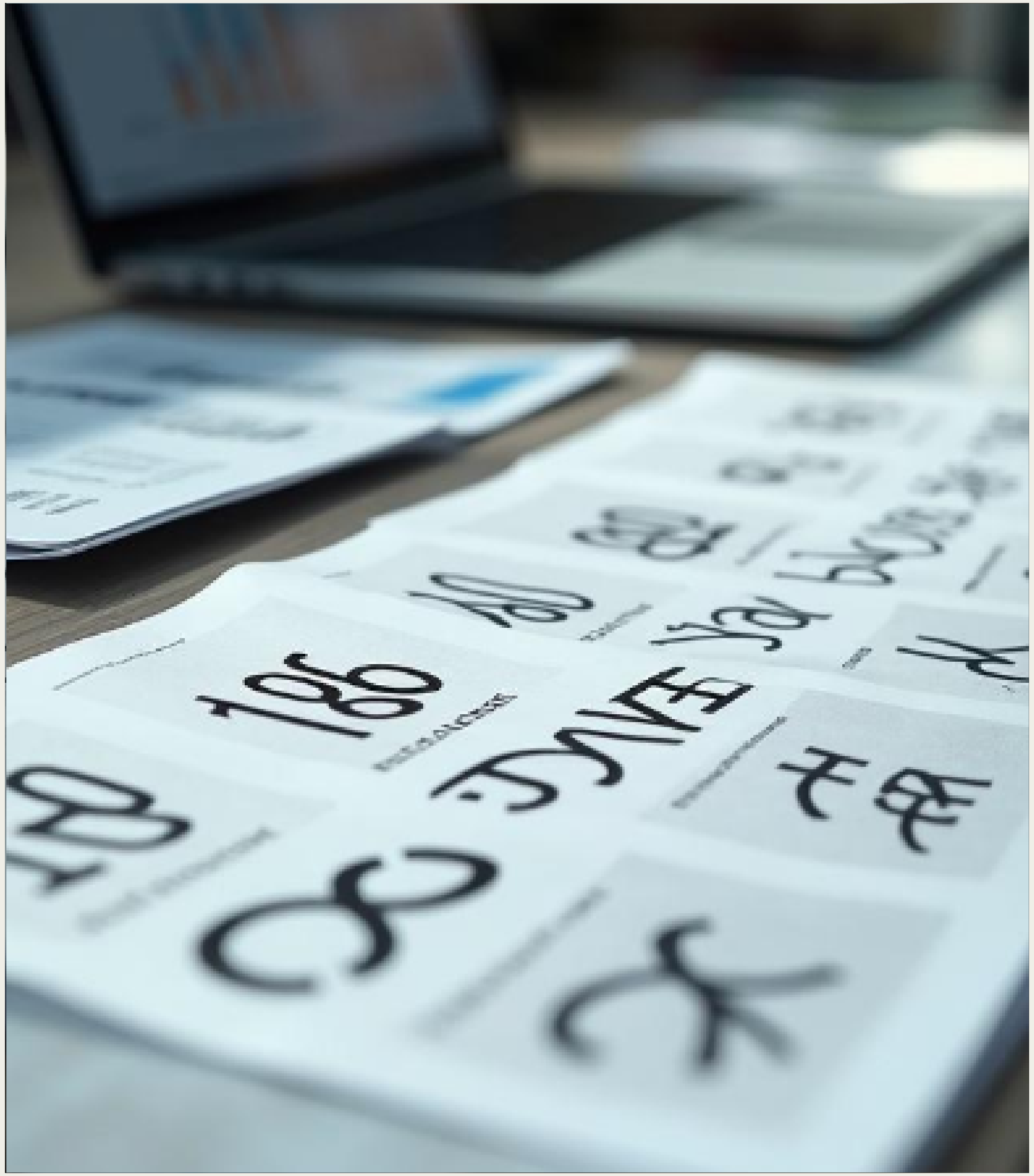
01

MNIST Dataset

Using tensor flow and keras

Overview of the dataset

The MNIST dataset is a large database of handwritten digits used for training various image processing systems. It consists of 60,000 training images and 10,000 testing images, all of which are grayscale and sized 28x28 pixels. Each image is associated with a label that indicates the digit it represents (0-9). Its simplicity and extensive use make it a benchmark dataset for machine learning models.



Data loading and preprocessing

Data loading involves obtaining the MNIST dataset using TensorFlow's built-in method, which divides the data into training and testing sets. Preprocessing includes normalizing pixel values to a range of 0 to 1 by dividing by 255.0, enhancing convergence during training. Additionally, reshaping the data for compatibility with convolutional neural networks is essential, transforming the images into a 4-dimensional array accommodating the batch size, height, width, and channels.

Normalization and reshaping

Normalization is a crucial step in machine learning to ensure that the input values are within the same range, which enhances the training process. In the case of the MNIST dataset, each pixel value is divided by 255.0 to scale it to a range of 0 to 1. This not only helps in faster convergence but also stabilizes the training process. Reshaping the images involves transforming the dataset from a 3D array into a 4D array, adjusting dimensions to include batch size and channel depth, which is essential for convolutional neural networks to interpret the data correctly.

02

CNN Model Development

Model architecture

The convolutional neural network (CNN) architecture for this task consists of several layers that progressively extract features from the input images. It starts with a convolutional layer that applies filters to detect edges and textures. This is followed by a max pooling layer to reduce dimensionality, allowing the model to focus on significant features. The network includes additional convolutional and pooling layers, culminating in fully connected layers that perform the classification into one of the ten digit categories.

Compilation and training

After defining the architecture, the model is compiled with an optimizer, loss function, and evaluation metrics. The Adam optimizer is typically used for its efficiency and effectiveness in handling sparse gradients. The loss function selected is sparse categorical cross-entropy, suitable for multi-class classification tasks. The model is then trained over multiple epochs on the training dataset, validating its performance on the test set to ensure that it generalizes well to unseen data.

Evaluation and predictions

Post-training, the model's performance is evaluated using the test dataset to assess its accuracy. The evaluation metrics, primarily accuracy, indicate how well the model recognizes handwritten digits. Additionally, predictions can be made on new data points, which can be visualized alongside the true labels for better understanding. This provides insights into how the model performs in practice and highlights areas for potential improvements.

Conclusions

In conclusion, the CNN model effectively demonstrates the process of recognizing handwritten digits from the MNIST dataset. Through careful normalization and reshaping of data, alongside a strategically designed architecture, the model achieves a high accuracy rate. The training, evaluation, and prediction phases collectively illustrate the capabilities of CNNs in image classification tasks. Future enhancements could explore advanced techniques such as data augmentation and deeper architectures for improved performance.

Thank you!

CREDITS: This presentation template was created by **Slidesgo**, and includes icons, infographics & images by **Freepik**

w
w
w.
y