**Project Report**

**Project Details**

* **Project Title:** Handwritten Digit Recognition using Convolutional Neural Networks (CNNs)
* **Project ID:** PRCP-1002-HandwrittenDigits
* **Project Team ID:** PTID-CDS-JAN-25-2385

**1. Introduction**

Handwritten digit recognition is a fundamental task in computer vision and pattern recognition. This project utilizes a Convolutional Neural Network (CNN) to classify handwritten digits from the MNIST dataset, a collection of 28×28 grayscale images representing digits from 0 to 9. The goal is to develop an efficient model capable of accurately recognizing digits and extending its applicability to custom handwritten digit images.

**2. Objectives**

* Implement a CNN model for digit recognition using TensorFlow/Keras.
* Train the model on the MNIST dataset and evaluate its performance.
* Test the trained model with unseen data and analyze its accuracy.
* Extend the model’s capability to recognize custom handwritten digits uploaded from Google Drive.

**3. Tools & Technologies Used**

* **Programming Language:** Python
* **Libraries:** TensorFlow, NumPy, Matplotlib, PIL (Python Imaging Library), Scikit-learn
* **Dataset:** MNIST (Modified National Institute of Standards and Technology)
* **Development Environment:** Google Colab

**4. Methodology**

**4.1 Data Preprocessing**

* Load the MNIST dataset using tf.keras.datasets.mnist.load\_data().
* Normalize pixel values to the range [0,1] by dividing each pixel by 255.0.
* Reshape images to add a single channel (grayscale) to match CNN input requirements.

**4.2 Model Architecture**

The CNN model is structured as follows:

1. **Conv2D** (32 filters, 3×3 kernel, ReLU activation, input shape: 28×28×1)
2. **MaxPooling2D** (2×2 pooling)
3. **Flatten** (Converts feature maps into a single vector)
4. **Dense** (64 neurons, ReLU activation)
5. **Dense** (10 neurons, Softmax activation for classification)

**4.3 Model Compilation & Training**

* The model is compiled using **Adam optimizer** and **sparse categorical crossentropy loss**.
* The model is trained using **15 epochs** with accuracy as the primary performance metric.

**4.4 Model Evaluation & Testing**

* The trained model is evaluated using the MNIST test dataset.
* Performance metrics include **accuracy, precision, recall, and F1-score** for comprehensive analysis.
* A sample test image is selected, and the model predicts its digit class.

**4.5 Custom Image Prediction**

* Google Drive is mounted to access user-uploaded images.
* The uploaded image is converted to grayscale and resized to 28×28 pixels.
* The image is normalized and reshaped before being passed to the trained CNN model.
* The model predicts the digit, and the result is displayed alongside the processed image.

**5. Results & Discussion**

* The model achieved an accuracy of **98.55%** on the MNIST test dataset.
* Additional evaluation metrics calculated using classification\_report from Scikit-learn:
  + **Precision:** Measures the proportion of true positive predictions for each digit class.
  + **Recall:** Measures the proportion of actual positives identified correctly.
  + **F1-score:** Harmonic mean of precision and recall for balanced evaluation.
* Predictions on test samples were highly accurate.
* The model performed well on custom handwritten images, demonstrating its generalization capability.
* Performance may vary based on the quality and clarity of handwritten digits in external images.

**6. Conclusion**

This project successfully implemented a CNN-based handwritten digit recognition system. The trained model demonstrated high accuracy on the MNIST dataset and performed well on custom handwritten images. The results indicate that CNNs are highly effective for digit classification and can be extended to more complex handwriting recognition tasks.