{AUTUMN INTERNSHIP PROJECT REPORT FORMAT}

Project Title

Image Transformations and MNIST Classification using PyTorch and TensorFlow-Keras

Student Name(s)

Sai Teja Ponnam

Course/batch: Autumn Internship 2025, ISI Kolkata

Period of Internship: 25th August 2025 – 19th September 2025

Report submitted to:
IDEAS – Institute of Data Engineering, Analytics and Science Foundation, ISI Kolkata

1. Abstract

In this project, I explored fundamental image processing and machine learning techniques. The work began with applying basic transformations such as grayscale conversion, shifting, and scaling on sample images using PIL and OpenCV. I then moved to classification tasks using the MNIST dataset of handwritten digits. First, a simple feedforward neural network was implemented using PyTorch. Then, a convolutional neural network (CNN) was developed using TensorFlow-Keras in an object-oriented style. Both models were trained and evaluated, and their performance was compared. This project helped me understand image preprocessing, model design, training pipelines, and how different frameworks (PyTorch and TensorFlow) implement deep learning tasks.

2. Introduction

Image classification is one of the key challenges in computer vision. The MNIST dataset, containing 70,000 handwritten digits (0–9), is widely used as a benchmark for testing machine learning models. In this project, I learned: - How to perform image manipulation using OpenCV and PIL - How to implement a feedforward neural network in PyTorch - How to design a CNN in TensorFlow-Keras using an object-oriented class - How to evaluate and compare models across frameworks The training during the first two weeks included Python programming, NumPy, Pandas, Matplotlib, Scikit-learn, PyTorch, TensorFlow, and OpenCV. This foundation was crucial for the project.

3. Project Objectives

- To perform basic image transformations (grayscale, shift, scale) using OpenCV and PIL. - To design and train a simple neural network in PyTorch for MNIST classification. - To implement a CNN in TensorFlow-Keras with an object-oriented approach. - To evaluate and compare results across frameworks. - To gain hands-on experience in data preprocessing, training, and evaluation pipelines.

4. Methodology

Image Transformations: - Converted color images (e.g., moon-pexels-frank-cone.jpg) into grayscale. - Compared grayscale conversion between PIL and OpenCV (different RGB weighting). - Performed shifting and scaling transformations using OpenCV and PIL. PyTorch Model: - Used the built-in MNIST dataset loader. - Built a feedforward network with dense layers and ReLU activation. - Trained using SGD optimizer for multiple epochs. - Evaluated on test data for accuracy. TensorFlow-Keras CNN: - Designed a CNN with two convolutional layers, max pooling, dense layers. - Used Adam optimizer and sparse categorical cross-entropy loss. - Wrapped the workflow into a class MNISTClassifier with methods for training and evaluation. Data Preprocessing: - Normalized pixel values to [0,1]. - Reshaped MNIST digits to (28, 28, 1) where required. - Used batch size = 64 and trained for 5 epochs. Tools & Libraries: - Google Colab / Jupyter - Python - Libraries: OpenCV, PIL, NumPy, Matplotlib, PyTorch, TensorFlow-Keras

5. Data Analysis and Results

- Image Transformations: - PIL and OpenCV grayscale images appeared different due to different conversion formulas. - Shifting and scaling operations were successfully demonstrated. - MNIST Classification: - PyTorch Feedforward Network → 95% accuracy on test set. - TensorFlow-Keras CNN → 98% accuracy on test set. - Result Comparison: - CNN outperformed the dense network since convolutional layers extract spatial features better.

6. Conclusion

This project gave me hands-on experience in image processing, machine learning, and deep learning. I learned how preprocessing affects model performance and how different frameworks can be used for the same task. - Feedforward neural networks are useful for simple tasks but have limited capacity. - CNNs capture spatial structure in images and hence perform better. Future Scope: Extending the project to datasets like CIFAR-10 or real-world images. Techniques such as data augmentation, dropout, and transfer learning can be applied to improve performance further.

7. Appendices

References: - PyTorch official documentation - TensorFlow/Keras documentation - OpenCV & PIL documentation GitHub Repository: https://github.com/saitejaponnam12/ISI_Autumn_Internship_Project_2025 Supporting Materials: - Notebook (.ipynb) - Report (.pdf/.docx) - Video demonstration (2 mins, mandatory) Datasets Used: - MNIST dataset (handwritten digits 0–9, available in PyTorch and Keras utilities) - Sample color image (moon-pexels-frank-cone.jpg) for transformation tasks