■ TechHorizon Internship – Deep Learning Projects

■ Objective:

This internship is designed to introduce you to deep learning through fun, practical projects. You will build simple neural networks using high-level libraries like Keras (TensorFlow) or PyTorch. The projects are curated to help you understand the basics of deep learning models and how they are applied in real-world tasks.

■ General Instructions:

- Complete all 3 tasks during the internship.
- Use clean, well-commented code in Jupyter Notebook or Python script.
- Include a short report (1–2 pages) summarizing: - Your approach - Challenges you faced - Key learnings and model performance

■ TASKS

Task 1: Build a Simple Neural Network to Classify Handwritten Digits

Goal: Use a basic neural network to classify digits from images.

Dataset: tensorflow.keras.datasets.mnist

- Load the MNIST dataset using Keras/TensorFlow.
- Normalize and flatten the image data.
- Build a simple neural network (2–3 dense layers).
- Train and test the model.
- Display a few predictions using matplotlib.

Task 2: Image Classification Using Pre-trained Models

Goal: Use a pre-trained model to classify general objects in images.

Dataset Suggestion: 10–15 sample images or Tiny ImageNet Subset

- Load a small number of sample images (can be your own or from a provided dataset).
- Use a pre-trained model like MobileNetV2 or ResNet50 from Keras.
- Resize images and use the model to predict image classes.
- Display predictions with images.

Task 3: Digit Drawing Recognition Web App (Optional Deployment)

Goal: Build a basic web interface where users draw digits and your model predicts them.

This task is optional but encouraged for those who want to learn basic model deployment.

- Use your model from Task 1.
- Build a simple web interface using Streamlit or Gradio.
- Let the user draw a digit and get prediction in real time.
- (Optional): Add a “Clear” or “Try Again” button.

■ Submission Guidelines

- Code: Submit clean and well-commented Python scripts or Jupyter Notebooks.
- Report: 1–2 pages summarizing your approach, challenges, and outcomes.
- Optional App: If you built the interface, share a link or how to run it locally.