**Road Sign Detection – Road Ragers**

**Deep Neural Networks Project**

Pavan Koushik Bonala – SE22UCSE326

Niyathi Vasasali – SE23LCSE005

Varun Utukuri - SE22UCSE285

This project implements and compares several neural network architectures (SimpleCNN, DeepFFN3, AdamNet, SimpleFFN) for detecting and classifying German traffic signs using the [GTSRB dataset](https://benchmark.ini.rub.de/gtsrb_news.html). All models are implemented from scratch in NumPy, with manual training loops, optimizers, and evaluation.

**Project Structure**

.

├── Models – Codes

└── Models/

├── SimpleFFN.py

├── DeepFFN3.py

├── AdamNet.py

├── SimpleCNN.py

├── Results – Jupyter Notebooks – contains the results – executed code

└── Models/

├── SimpleFFN.ipynb

├── DeepFFN3.ipynb

├── AdamNet.ipynb

├── SimpleCNN.ipynb

├── Road Ragers Report.pdf

└── Dataset/

├── Dataset.zip

**Dataset/Train/**: Contains 43 subfolders (0–42), each with images for that class.

**Each \*model\_name.py**: Script for a specific model.

**Each .ipynb**: Contains the Neural Network models executed with results

**Each script includes:**

* Data loading and preprocessing
* Model definition
* Training and validation with hyperparameter tuning
* Accuracy/loss plotting

**Usage**

1. **Prepare the dataset:**

Download the [GTSRB dataset](https://benchmark.ini.rub.de/gtsrb_news.html).

Place the training images in **Dataset/Train/0**, **Dataset/Train/1**, ..., **Dataset/Train/42**.

1. **Install requirements:**

bash pip install numpy scikit-learn matplotlib pillow opencv-python

1. **Run a model script:**  bash python SimpleCNN.py
2. **View results:** Training and validation accuracy/loss will be printed and plotted.