Weekly Project: Traffic-Related Deep Learning Project

Objective:

Students will choose a traffic-related project and implement a deep learning model to address the problem. This project will allow students to apply their knowledge of neural networks, including the selection and implementation of the most appropriate model.

Project Overview:

The goal is to create a model that can effectively solve a traffic-related problem (e.g., traffic sign recognition, vehicle detection, traffic flow analysis) using a chosen dataset. Students will be responsible for the entire workflow, from data collection and preprocessing to model selection, implementation, and evaluation.

Project Tasks:

- **Task 1: Project Idea Selection and Dataset Collection**
  - Choose a traffic-related project (e.g., traffic sign recognition, vehicle detection, traffic flow analysis).
  - Gather the data either by obtaining an existing dataset or by scraping data online.
  - Store the collected data in a database (e.g., SQLite, MongoDB).
  - Explore the dataset and understand its structure.
  - Perform data preprocessing steps such as resizing, normalization, and data augmentation.
  - Split the dataset into training, validation, and test sets.

- **Task 2: Model Selection and Implementation**
  - Based on your project requirements, select the most suitable deep learning model (e.g., CNN, ANN, or any other deep learning model).
  - Design and implement the selected model using a deep learning framework such as TensorFlow or PyTorch.
  - Train the model using the prepared dataset.
  - Fine-tune hyperparameters such as learning rate, batch size, and the number of epochs.
  - Track model performance on the validation set during training.

- **Task 3: Model Evaluation and Reporting**
    - Evaluate the model on the test set using performance metrics relevant to your project.
    - Compile all findings, code snippets, and results into a final report.
    - Prepare a presentation summarizing the project, including the problem statement, methodology, results, and conclusions.

Deliverable: A notebook with the entire workflow (data collection, preprocessing, model selection, implementation, training, evaluation), a presentation slides.

Team Collaboration and Task Assignment:

Each student should contribute to the project, and it is important to document who did each task. Ensure that all tasks, from data gathering and preprocessing to model design and evaluation, are clearly assigned and recorded in the presentation.

Presentation Questions:

To assist with the presentation, consider the following questions:

1. What challenges did you face during data collection and how did you overcome them?
2. Why did you choose your specific model architecture?
3. How does the performance of your chosen model compare to alternative approaches, and what factors contributed to the differences?
4. What did you learn from storing the data in a database, and how did it impact your workflow?
5. If you were to extend this project, what additional features or improvements would you consider?

Main Grading System:

The following table outlines the grading criteria for the main project. Each component is assigned a maximum number of points, with the total possible score being 100 points. The grade then will be transformed into 20 grades on this week's evaluation.

Component	Description	Points
Project Idea Selection	Clarity, feasibility, and relevance of the chosen traffic-related problem.	15
Data Collection & Preprocessing	Quality of data collection, handling missing data, normalization, and augmentation.	20
Model Selection & Implementation	Appropriateness of model choice, design, and implementation, including code quality and efficiency.	20
Model Training & Evaluation	Effectiveness of model training, hyperparameter tuning, and evaluation on test data.	20
Database Integration	Proper storage and retrieval of data from a database, including documentation of the process.	10
Presentation	Quality of the presentation, including clear explanation of the problem, methodology, and results.	15
Team Collaboration & Contribution	Equal participation, clear task assignment, and documentation of who did each task.	10

Bonus Tasks for Extra Credit:

Students could complete the following bonus tasks to earn extra points if they missed grades in the main project. These tasks encourage additional effort, creativity, and the use of advanced techniques.

Bonus Task	Description	Bonus Points
Data Scraping or Manual Data Collection	If students scraped data from the web or collected data manually instead of using an existing dataset.	+10
Use of Advanced Frameworks (e.g., PyTorch)	If students used advanced frameworks like PyTorch instead of simpler tools.	+5
Implementation of Custom Data Augmentation	If students implemented custom data augmentation techniques beyond standard transformations.	+5
Advanced Model Architectures	If students implemented advanced model architectures instead of simpler models.	+10
Hyperparameter Tuning with Grid/Random Search	If students applied hyperparameter tuning methods like Grid Search or Random Search to optimize their model.	+5
Ensemble Learning	If students implemented ensemble methods to improve the model's performance.	+10
Oversampling or Undersampling Techniques	If students applied oversampling (e.g., SMOTE) or undersampling techniques to handle imbalanced datasets effectively.	+5
Deployment of the Model	If students deployed their model on a cloud service or as a web application/API.	+20