

ChatGPT

Here's a fresh summary and presentation pitch based on your document:

Good morning, everyone. Today, we are excited to present our project on **Traffic Sign Classification using Convolutional Neural Networks**, developed as part of the **Digital Image Processing Lab (CSEL-4104)** under the guidance of **Dr. Md. Zulfiker Mahmud**. This project represents our exploration of deep learning technologies applied to traffic sign recognition.

Objective

Our primary objective was to create a system that can accurately recognize and classify traffic signs. This involves a combination of **Digital Image Processing (DIP)** and **Convolutional Neural Networks (CNNs)** to achieve a robust and effective classification model. Such a system is essential for autonomous driving, smart driving assistance, and overall traffic safety enhancement.

Motivation

The motivation for this project lies in the potential to make road navigation safer and more efficient. By developing a reliable traffic sign recognition system, we aim to assist both drivers and pedestrians, enabling on-the-go identification of road signs and supporting autonomous vehicle technologies.

Image Acquisition & Dataset

For our training, we utilized the **German Traffic Sign Recognition Benchmark (GTSRB)** dataset, which contains over **35,000 images**, each sized **32x32 pixels**, spanning **43 classes** of traffic signs. This dataset provides the diversity needed to train a comprehensive model capable of handling various types of signs.

Data Preprocessing and Augmentation

To ensure optimal model performance, we:

- Converted images to **grayscale** for simplified processing.
- Applied **histogram equalization** to enhance contrast.

- **Normalized** data and scaled pixel values to a range of **[0,1]** for consistent inputs.

For better model generalization, we employed **data augmentation techniques** such as width and height shifting, zooming, shearing, and rotation. These steps generated variations in our dataset to prevent overfitting and ensure adaptability.

Model Structure and Training

Our CNN model is designed with:

- **Convolution layers** for feature extraction.
- **Pooling layers** for dimensionality reduction.
- **Dropout layers** to prevent overfitting.
- **Dense layers** for feature integration.
- **ReLU** and **Softmax** functions for activation and classification outputs.

Training was conducted with an **80/20 split** between the training and validation sets, using a **batch size of 2000** and **10 epochs**. We chose the **Adam optimizer** for its efficiency in model convergence.

Results and Live Testing

Our model demonstrated strong performance, achieving a **test accuracy of 98.29%** with a corresponding test loss of **6.19%**. Live testing further confirmed the model's reliability, showcasing real-time recognition capabilities.

Deployment

To make this solution practical, we developed:

1. An **Android app** utilizing **TensorFlow Lite, OpenCV, Java, and CameraX** for on-the-go traffic sign recognition via the camera.
2. A **web-based application** built with **Flask, HTML, CSS, and JavaScript** to allow users to classify traffic signs conveniently online.

Future Work

Looking ahead, we plan to:

- Develop a dataset with **Bangladeshi traffic signs** for enhanced local applicability.
- Integrate **audio alerts** for detected traffic signs.

- Implement **augmented reality (AR)** features for real-time visual displays.
-

Thank you for your attention. We're happy to answer any questions you may have.

Let me know if you'd like any additional details or modifications!