Traffic Sign Recognition
With Audio Alert

Group 1

Jeet Parekh
Sally wei

Lakshmi Satvika Nekkanti

Shashidhar Reddy

Motivation & Background

- Road sign detection is a crucial task in the field of autonomous driving and intelligent transportation systems.
- This Enables vehicles to understand the Environment and make informed decisions based on traffic signs.
- In extreme weather conditions such as heavy rain and fog audio based systems can help drivers recognize and respond to road signs.
- By providing an audio alert, the system can ensure that drivers are aware of the information displayed on the signs.
- Incorporating audio in road sign detection systems can enhance the accessibility of these systems, and contribute to safer driving experiences.

Literature Review

Paper	Model	Accuracy
Dewi et al. (2021)	DCGAN	92%
Shivayogi et al. (2022)	YOLOv4	54.68–76.55%
Sukhani et al. (2021)	GoogleNet, VSSA, VGGNet	80.5–98.52%
Xie et al. (2022)	Federated & non-federated SNN	~95.8%

- Detection time
- Limited energy and computing resources

Methodology

Dataset Description : A german traffic sign dataset which has 51839 images and 44 classes with different traffic signs.

The dataset is initially in the size of (256,256) and then later resized it to (30,30).

Software Requirements :

- Libraries : Tensorflow,Keras , PIL for converting the images into array of numbers , pandas, numpy ,matplotlib and tensorflow, Pyttsx3 a python library to convert from text to speech.
- IDE : VSCode , Programming language :Python
- CUDA : To train the model on a GPU

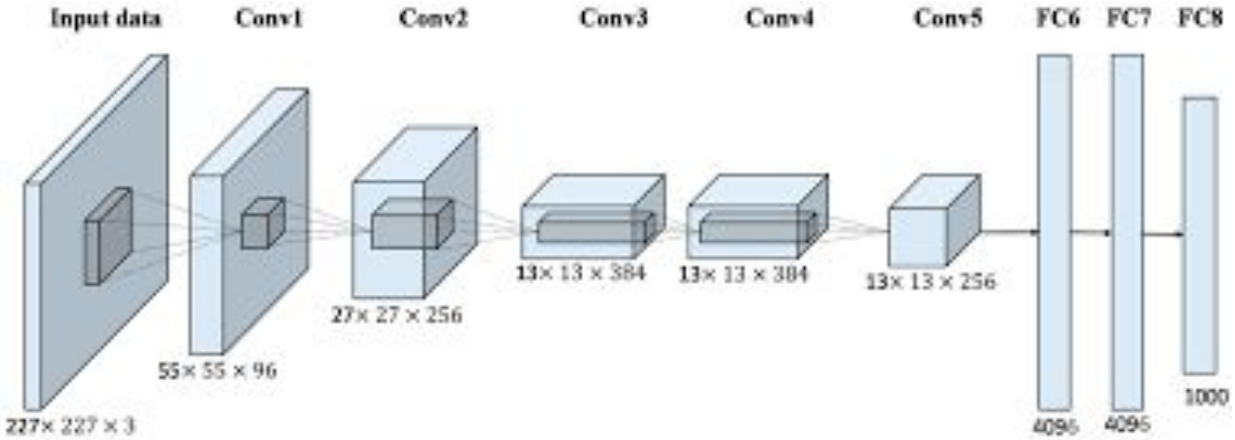
Models Used :

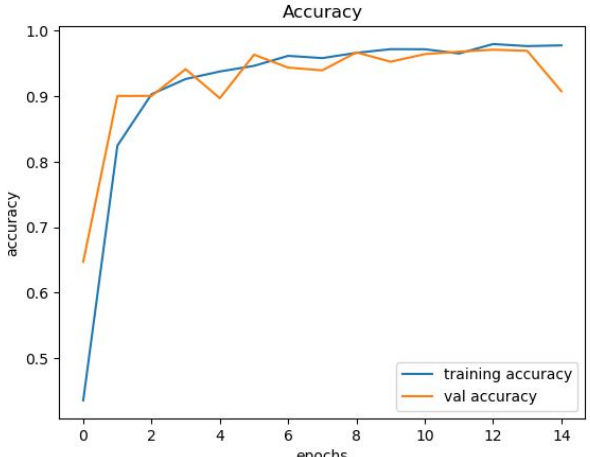
- Alex Net
- CNN
- VGG-19

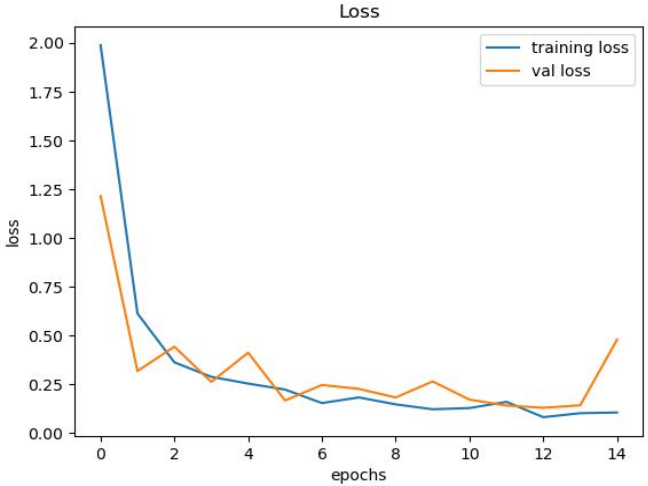
Pre-Processing

- Here we aim to enhance the training results of the dataset by reducing noise in the image data.
- The dataset contains image files and metadata files that provide coordinates to locate traffic signs within the images.
- Libraries such as pandas, numpy, and OpenCV (cv2) are used for data handling, image reading, manipulation, and saving.
- Cropping images based on coordinates helps in Edge detection and in removal of irrelevant parts and focus on the traffic signs, improving the quality of the training data.
- By separating the preprocessed images into a distinct output directory, we have keeps them separate from the original dataset, ensuring organization and clarity.
- The image retrieval process involves resizing the images to a standardized size of (30,30) to ensure consistency and converting the images into numpy arrays for further processing.

Alex Net





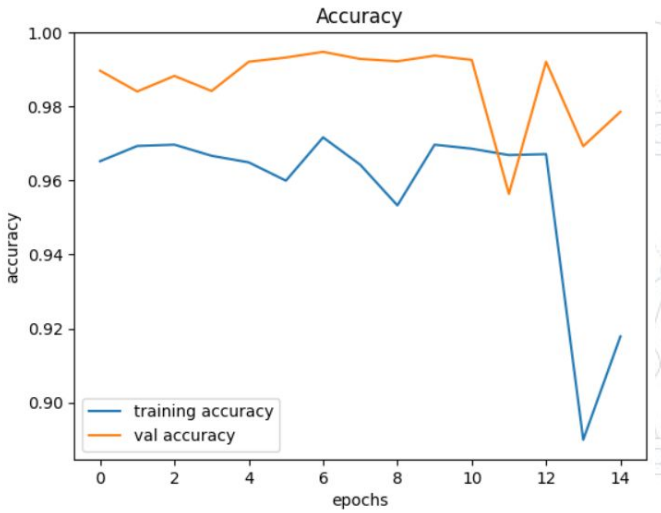


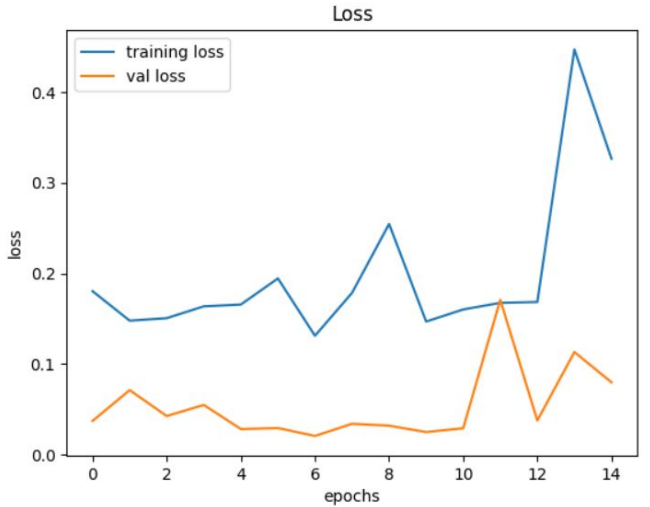
Accuracy	96%
Epochs	15
Loss	0.1045
Optimizer	Adam
Learning Rate for adam optimizer	0.001
Learnable	21,666,957

Custom CNN

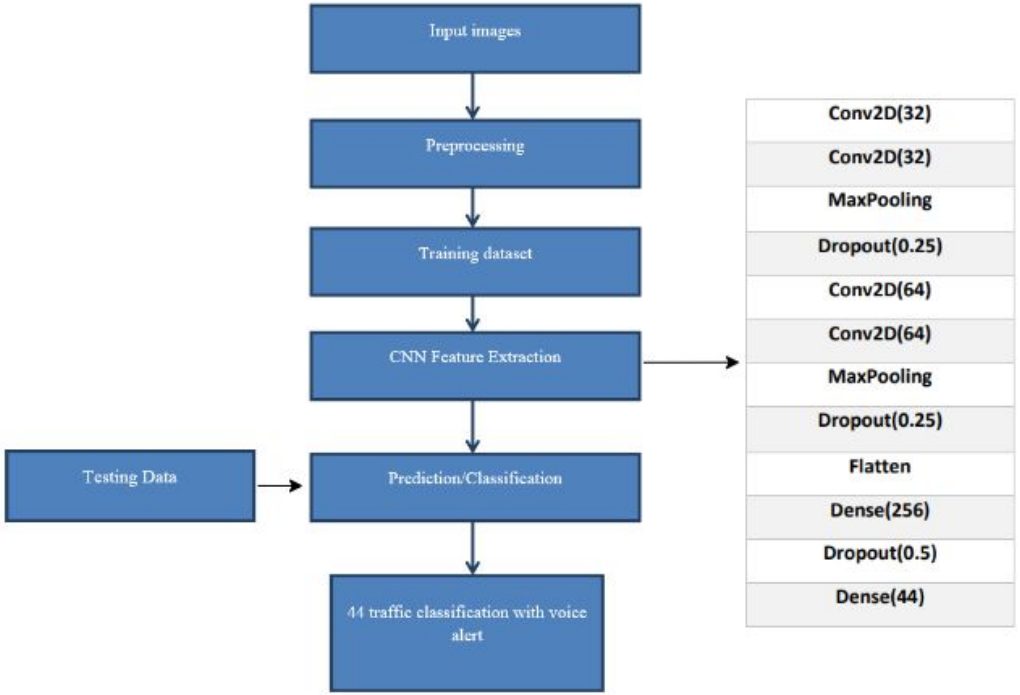
- Developed a custom Convolutional Neural Network (CNN) model for traffic sign classification.
- Constructed the CNN architecture with Conv2D, MaxPool2D, Dropout, and Dense layers.
- Incorporated activation functions, such as ReLU, to introduce non-linearity in the model.
- Configured the model with appropriate activation functions and input/output dimensions.
- Compiled the model with the categorical cross-entropy loss function and the Adam optimizer.

Accuracy	97.85%
Loss	0.079
Optimizer	Adam
Learnable parameters	242,508





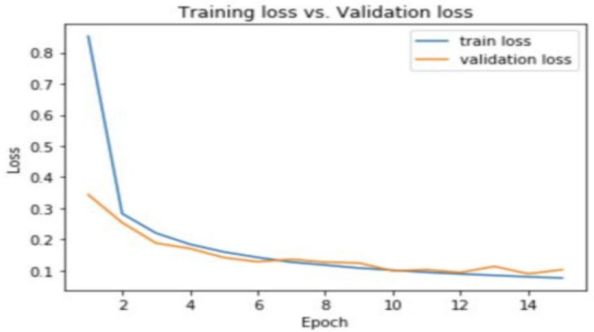
Custom CNN Architecture

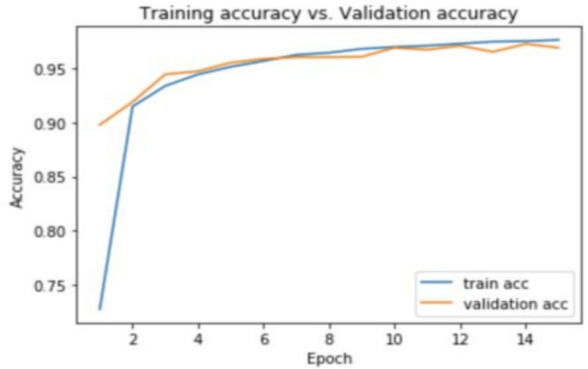


VGG19

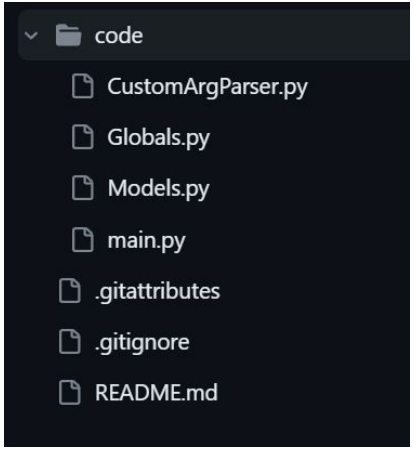
- The VGG19 is a convolutional Neural network which has 19 layers, including 16 CNN layers and 3 fully connected layers.
- The output of each CNN is passed through a ReLU activation function.
- Pros:It uses multiple CNN layers with small filter sizes, which helps network to learn more complex features.

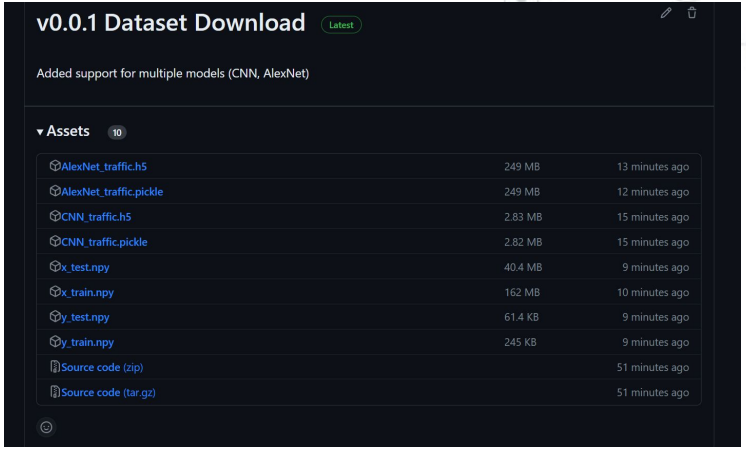
Accuracy	95%
Loss	1.75
Learnable parameters	21,766,956





CLI






```
usage: main [-h] [-t] [-r] [-s] [-m MODEL]

Traffic Sign Recognition.

optional arguments:
  -h, --help            show this help message and exit
  -t, --train           download training data set and train the model
  -r, --results         show results after validation and testing
  -s, --split           download the raw data, split it into training and test, and save to file
  -m MODEL, --model MODEL
                        Specify which model to run. If unspecified, defaults to CNN. Valid options: CNN, AlexNet

Dataset: The German Traffic Sign Recognition Benchmark https://benchmark.ini.rub.de/gtsrb_news.html
```

```
PS DATA255_Project> py code/main.py -t -m 'AlexNet' -e 1
Selected model: AlexNet
Training for custom number of epochs: 1
Training data loaded from ..\training\
 863/1961 [============>.................] - ETA: 5:09 - loss: 2.1917 - accuracy: 0.3765
```

Conclusion

- Implemented the Traffic Sign Board Detection and Voice Alert System using Convolutional Neural Network (CNN).
- Explored various CNN models and selected the one with the highest accuracy on the GTSRB dataset.
- Created separate classes for each traffic sign, improving the model's accuracy.
- Integrated voice alert functionality to provide real-time alerts to the driver upon sign recognition.
- Significantly advances the field of driving by enhancing driver convenience without compromising safety.
- Offers a scalable and accessible solution as it requires minimal hardware implementation.

Future Scope

- CLI: Adding ability to select which image to test the model on
- GUI: Adding ability to select which model to run
- Testing additional models
- Adding multiple languages to the speech

Contribution

- Shashidhar : Dataset,VGG19
- Lakshmi Satvika:AlexNet and speech output using pyttsx3 (text to speech conversion library).
- Jeet :GUI, CNN
- Sally: CLI