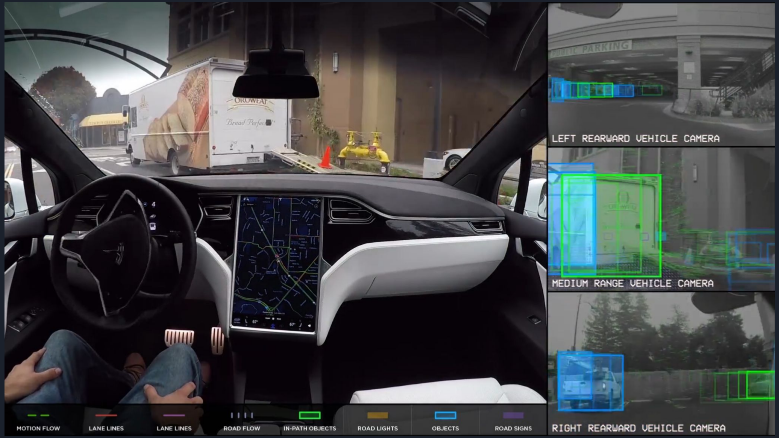
Traffic Sign Recognition using Deep Learning

Zaid Khan Final Project Presentation

Overview of Traffic Sign Recognition

- Recognizing traffic signs is crucial for autonomous driving.

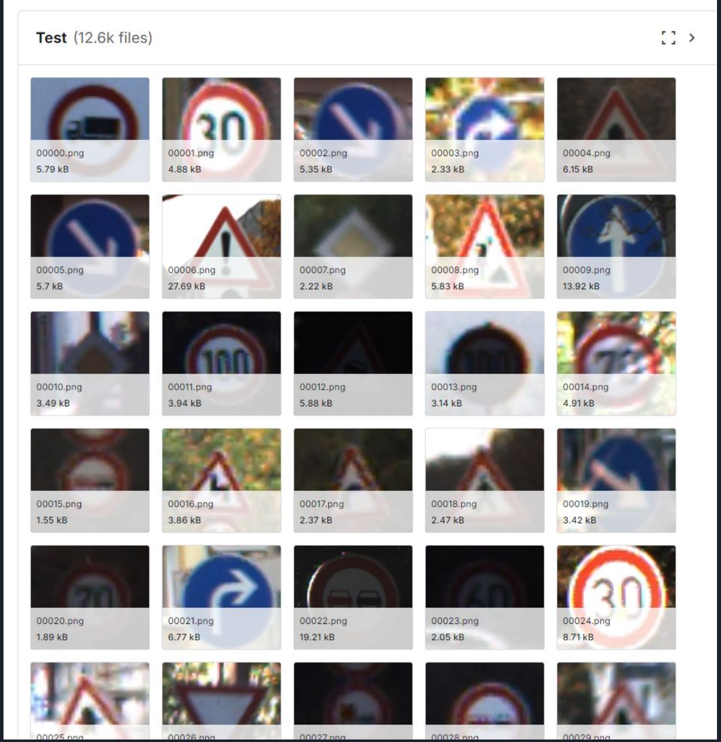
- Real-time detection and classification enhance safety for automated and driver-assist systems.


LEFT REARWARD VEHICLE CAMERA
MEDIUM RANGE VEHICLE CAMERA
RIGHT REARWARD VEHICLE CAMERA
MOTION FLOW
LANE LINES
LANE LINES
ROAD FLOW
IN-PATH OBJECTS
ROAD LIGHTS
OBJECTS
ROAD SIGNS

https://www.tesla.com/autopilot

Why?

- ➢ Relevance to real-world applications in autonomous vehicles
  - ○ *tesla please stop rejecting my internship apps.*
- ➢ Opportunity to learn more about CNN architecture and live implementation.
- ➢ Challenge of working with a dataset with multiple classes and varied conditions.
- ➢ The German Traffic Sign Recognition Benchmark (GTSRB) already exists as a dataset on kaggle.



Test (12.6k files)

00000.png 5.79 kB
00001.png 4.88 kB
00002.png 5.35 kB
00003.png 2.33 kB
00004.png 6.15 kB
00005.png 5.7 kB
00006.png 27.69 kB
00007.png 2.22 kB
00008.png 5.83 kB
00009.png 13.92 kB
00010.png 3.49 kB
00011.png 3.94 kB
00012.png 5.88 kB
00013.png 3.14 kB
00014.png 4.91 kB
00015.png 1.55 kB
00016.png 3.86 kB
00017.png 2.37 kB
00018.png 2.47 kB
00019.png 3.42 kB
00020.png 1.89 kB
00021.png 6.77 kB
00022.png 19.21 kB
00023.png 2.05 kB
00024.png 8.71 kB
00025.png
00026.png
00027.png
00028.png
00029.png

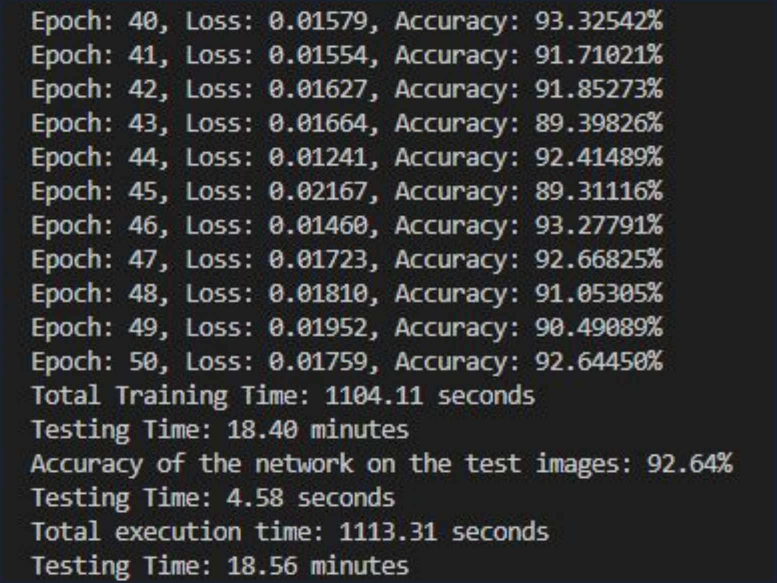
[GTSRB Dataset on Kaggle]

The Model

- Libraries and Dataset:
  - Torch, Torchvision
  - Dataset: GTSRB, consisting of 43 traffic sign classes.
- Model Architecture:
  - Using ResNet50 as my base case for the model.
  - Using ResNet-152 to experiment between the two.
  - Adjusting the final layer for 43-class classification.

Training

- Training Loop:
  - Model parameters are optimized using Adam and cross-entropy loss.
  - Accuracy and loss computed for both training and validation sets each epoch.
- Hyperparameters:
  - Batch size: 32, Epochs: 50, Learning Rate: 0.001.
- Used RTX 4060ti for cuda processing

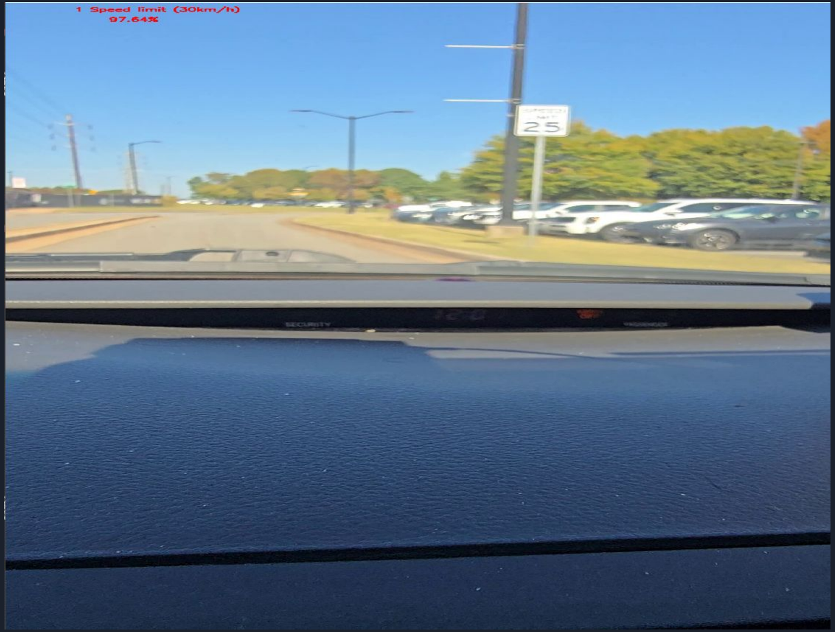


```
Epoch: 40, Loss: 0.01579, Accuracy: 93.32542%
Epoch: 41, Loss: 0.01554, Accuracy: 91.71021%
Epoch: 42, Loss: 0.01627, Accuracy: 91.85273%
Epoch: 43, Loss: 0.01664, Accuracy: 89.39826%
Epoch: 44, Loss: 0.01241, Accuracy: 92.41489%
Epoch: 45, Loss: 0.02167, Accuracy: 89.31116%
Epoch: 46, Loss: 0.01460, Accuracy: 93.27791%
Epoch: 47, Loss: 0.01723, Accuracy: 92.66825%
Epoch: 48, Loss: 0.01810, Accuracy: 91.05305%
Epoch: 49, Loss: 0.01952, Accuracy: 90.49089%
Epoch: 50, Loss: 0.01759, Accuracy: 92.64450%
Total Training Time: 1104.11 seconds
Testing Time: 18.40 minutes
Accuracy of the network on the test images: 92.64%
Testing Time: 4.58 seconds
Total execution time: 1113.31 seconds
Testing Time: 18.56 minutes
```

Real Time Recognition

- Video Capture and Preprocessing:
  - Capturing frames from a video file, resizing, and transforming each frame for model input.

- Inference:
  - Softmax to obtain probability for each class, with thresholding to determine detection confidence.
  - Displaying results on video frames in real-time.



Challenges

- Noise
  - The video I had was very blurry because my camera kept refocusing on my dashboard

- General model accuracy on the video feed
  - The model was trained on a foreign dataset
  - Different batch sizes, models, and general hyperparameters could be affecting the classification
  - The recognition model looks at the entire image not at the specific signs

Future work

- Better video for testing
- Create a detection algorithm to detect where the traffic signs are in the video
- More localized datasets
  - https://www.kaggle.com/datasets/pkdarabi/cardetection
  - https://git-disl.github.io/GTDLBench/datasets/lisa_traffic_sign_dataset/
- Finally move on to flags and racetrack machine vision classifications