BOSCH'S TRAFFIC SIGN RECOGNITION

—

INTER IIT TECH MEET ‘21

Introduction

Deep learning has proved to be a great tool in solving many tasks in image recognition that was once considered impossible for humans. However, it still isn't easy for a beginner in the field to analyse the results and curate a dataset to ensure that the AI is exposed to more difficult images that are likely to be seen in a real-world scenario. The solution we developed provides one such tool in the form of a UI that enables users to create difficult datasets for rigorous testing of the model, quickly analyse results, and find shortcomings in the dataset and the model.

Description

We have developed an end to end solution that enables a user, with the help of an intuitive UI to

a. Create a complex Dataset
b. Train additional images on the fly
c. View model performances across different metrics
d. Visualize model performance
e. Get suggestions to various shortcomings in model training
f. An explainable AI-based solution to comprehend network failures

Method:

The entire training paradigm for all ML models used is built in python using DL libraries like Pytorch and Tensorflow. The UI frontend is made using React and integrated with the backend using Flask API.

Dataset creation method

The original dataset given (GTSRB) contains 43 classes with 40,490 images.

One of our main tasks was to make this dataset more difficult. To achieve this we added 7 more classes to the existing 43 classes. The newly added classes are as follows:

- No stopping
- Cross-road
- No passing cars
- Route for pedal vehicles only
- Separated tracks for pedal cyclists and pedestrians only
- Parking sign
- Tonnage limit

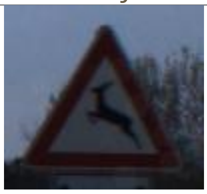
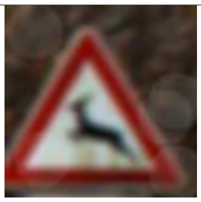
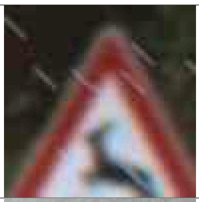
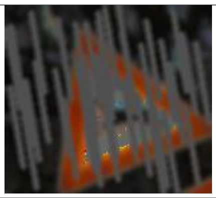
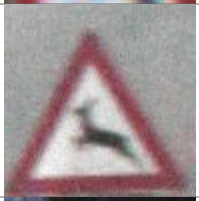
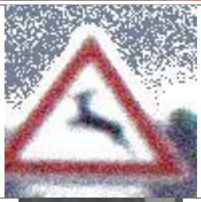
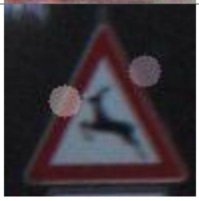
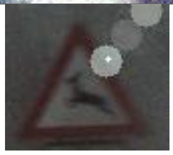
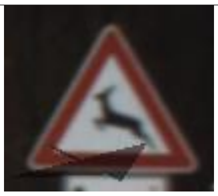
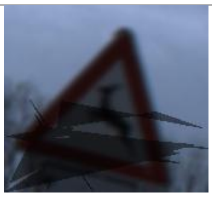
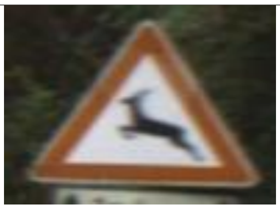
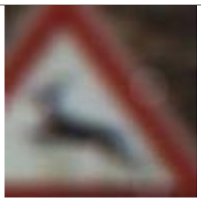
This dataset (now 50 classes) was not very difficult and our various models achieved near perfect results which are summarized as below:

The baseline model was a simple 4 conv layer model which was just used as a tool to understand the complexity of the model required for the dataset.

	Accuracy	**Precision**	**Recall**	**F1 Score**
Baseline	0.9906	0.9925	0.9889	0.9907
Baseline augmented	0.8001	**0.9584**	0.7272	0.8170
Resnet50	0.8359	0.9237	0.8072	0.8613
MobilenetV3	0.886	0.9535	0.8631	0.9058
InceptionV3	**0.8878**	0.9567	**0.8675**	**0.9097**

Further increasing the difficulty

We used a variety of augmentations on our modified dataset to make it even more difficult. We used augmentations like center-crop to add some images with partial information. The list of augmentations is as follows:

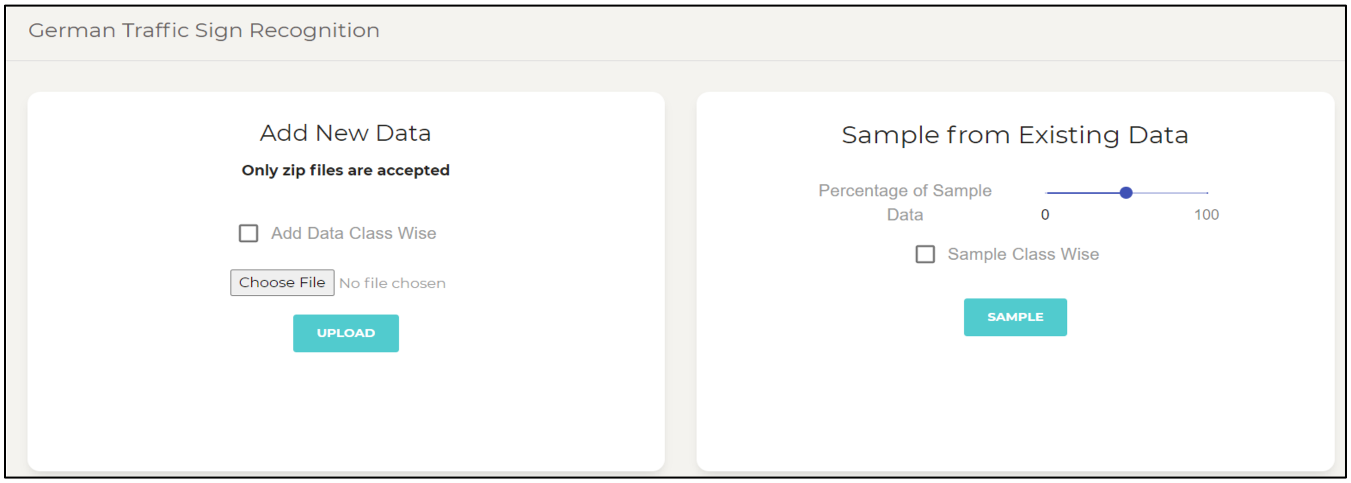
Augmentation	Easy	Hard
Random Fog		
Random rain		
Random Snow		
Random Sun Flare		
Random Shadow		
Gaussian Noise		

Augmentation	Easy	Hard
Random Fog	Blur Limit=3	Blur Limit=6
Random Rain	Rain Type= Drizzle	Rain Type= Torrential
Random Snow	Brightness Coefficient=2.5	Brightness Coefficient=5
Random Sun Flare	Sun Flare Radius= 10	Sun Flare Radius= 25
Random Shadow	No. Shadow Lower=2 No. Shadow Upper=3	No. Shadow Lower=3 No. Shadow Upper=6
Gaussian Noise	Variance limit=(10.0, 20.0)	Variance limit=(10.0, 40.0)

To ensure the high level of difficulty of our dataset, we used a Hard level of augmentations. Our final dataset had **1,74,952** numbers of images.

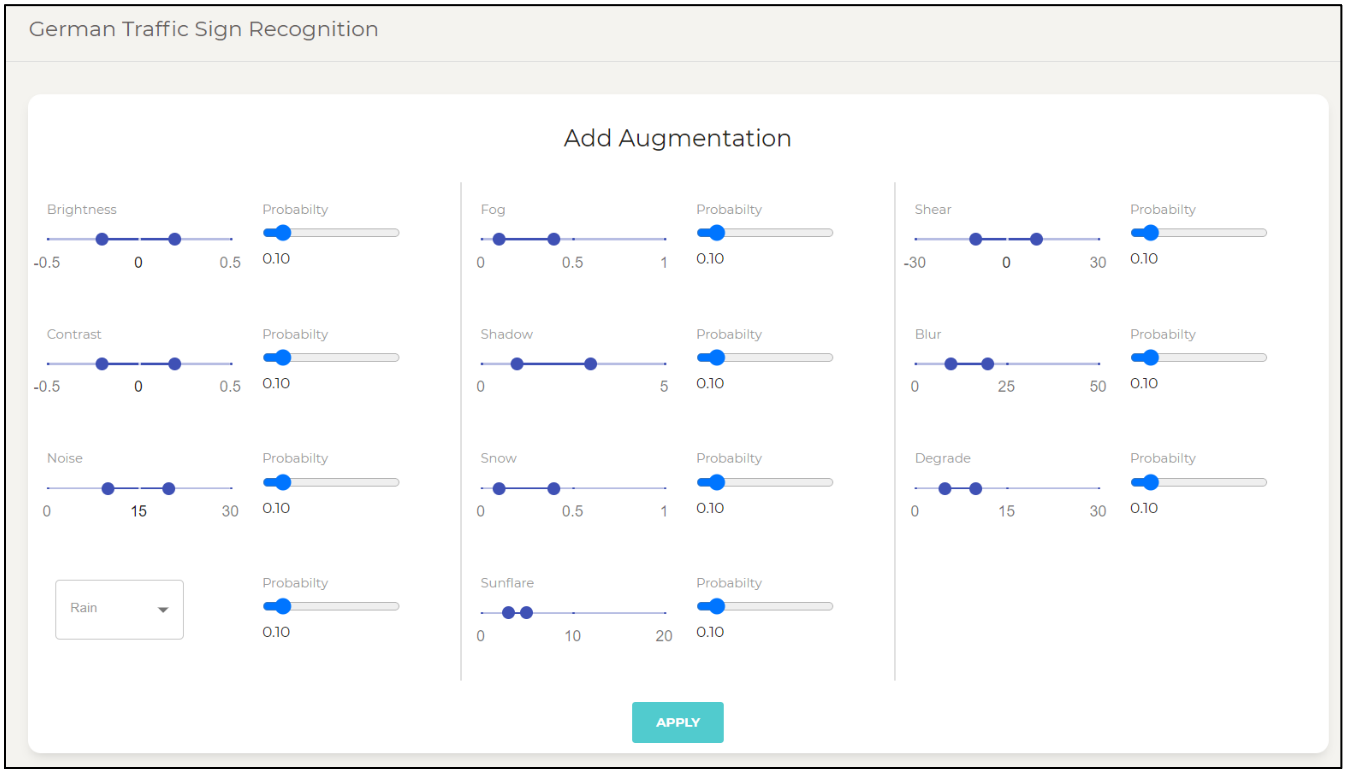
Upload dataset

- For Data Input, the user is provided with two options of either uploading new images (only zip files) or uploading data belonging to a particular class
- Further, the user can also sample from existing data with a desired percentage.
- Finally, the user can visualize the newly added data.


German Traffic Sign Recognition
Add New Data
Only zip files are accepted
Add Data Class Wise
Choose File
No file chosen
UPLOAD
Sample from Existing Data
Percentage of Sample Data
0
100
Sample Class Wise
SAMPLE

Add augmentations

We allow the user to choose from **13** different types of augmentations and apply them to the uploaded images using an easy-to-use UI and make the dataset more difficult.



The user can specify the min and max range for the augmentations and the probability with which that augmentation would be applied to the dataset.

Users can visualize the augmented dataset and remove images as he/she see fit.

Training

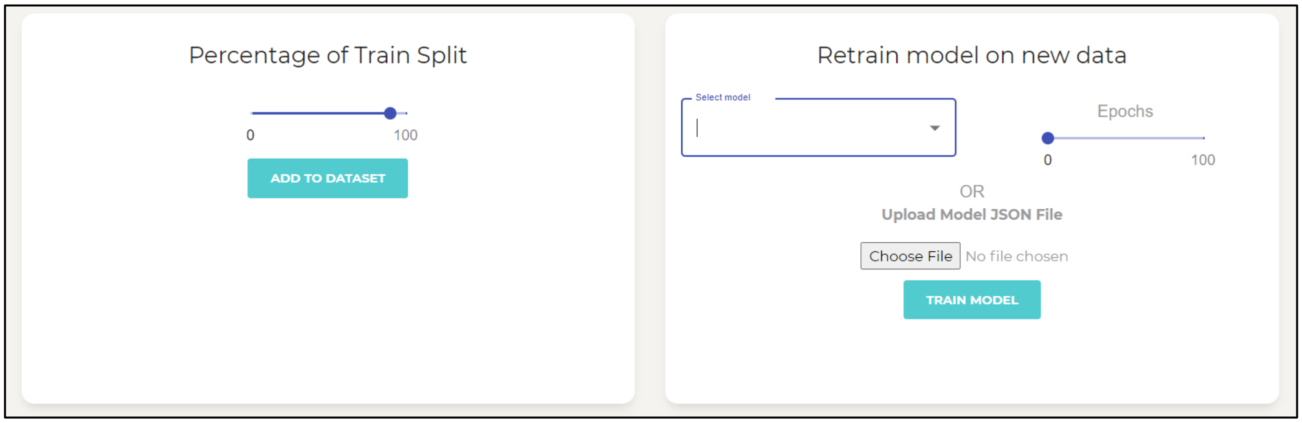
Once the user is satisfied with the level of difficulty of the dataset he/she can then split the dataset and train the model from a list of pre-trained models available on our platform.

The train-test split follows stratified sampling wherein the class ratio is maintained in both train and validation dataset.

Available pre-trained models:

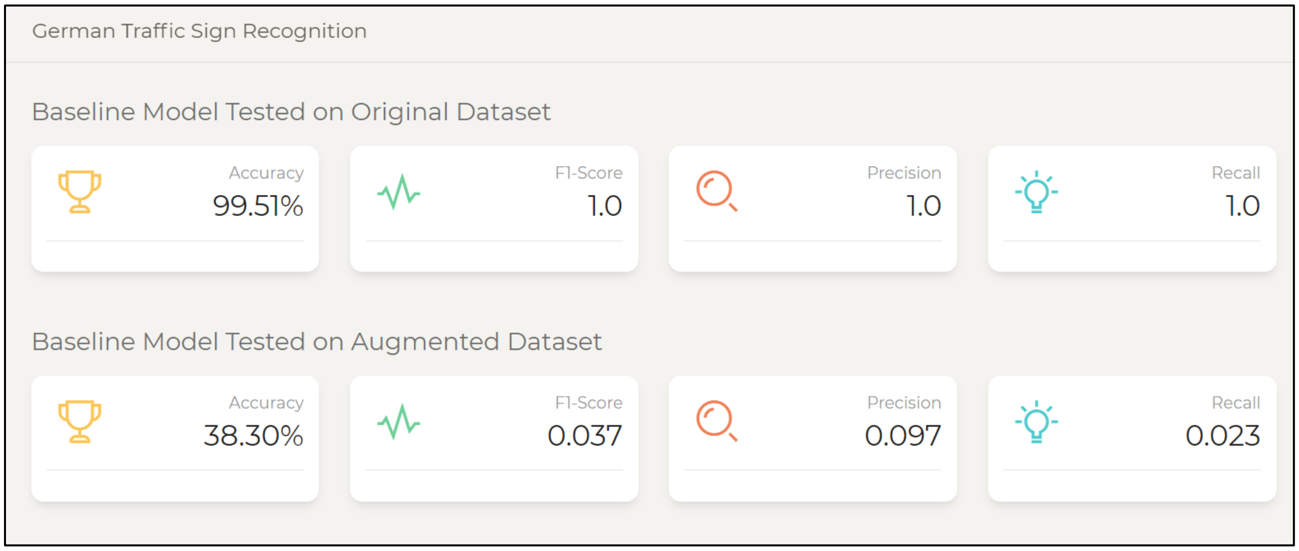
1. Baseline
2. Baseline Augmented
3. InceptionV3
4. MobileNetV2
5. MobileNetV3

We also provide an option to upload custom model JSON file which could be trained from scratch.



Model performance and statistics

A dashboard that shows accuracy on the base dataset and an augmented much harder dataset for any model selected is always shown.



German Traffic Sign Recognition

Baseline Model Tested on Original Dataset

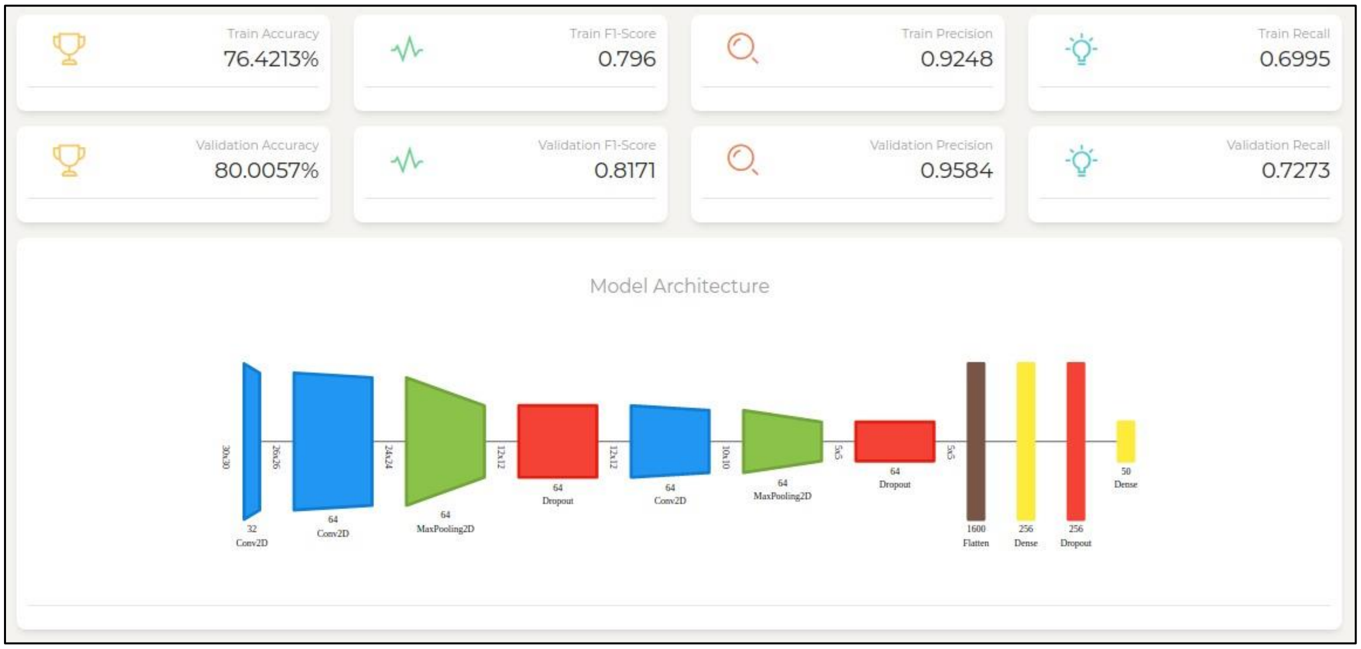
Accuracy	F1-Score	Precision	Recall
99.51%	1.0	1.0	1.0

Baseline Model Tested on Augmented Dataset

Accuracy	F1-Score	Precision	Recall
38.30%	0.037	0.097	0.023

The user can choose the model from the dropdown for which the statistics needs to be displayed.

In addition, testing on any new data added to our already augmented dataset will result in the following metrics being displayed in the UI and corresponding to every model pre-trained or custom we have a tab to display model architecture.



Train Accuracy 76.4213%

Train F1-Score 0.796

Train Precision 0.9248

Train Recall 0.6995

Validation Accuracy 80.0057%

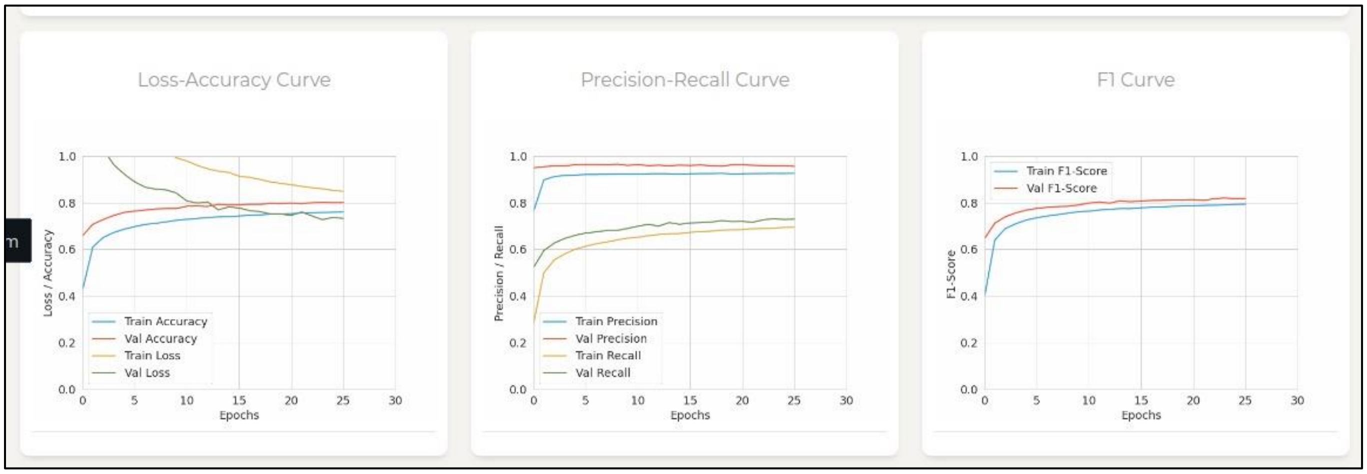
Validation F1-Score 0.8171

Validation Precision 0.9584

Validation Recall 0.7273

Model Architecture

Along with this, the loss, accuracy and F1 curve can be seen.

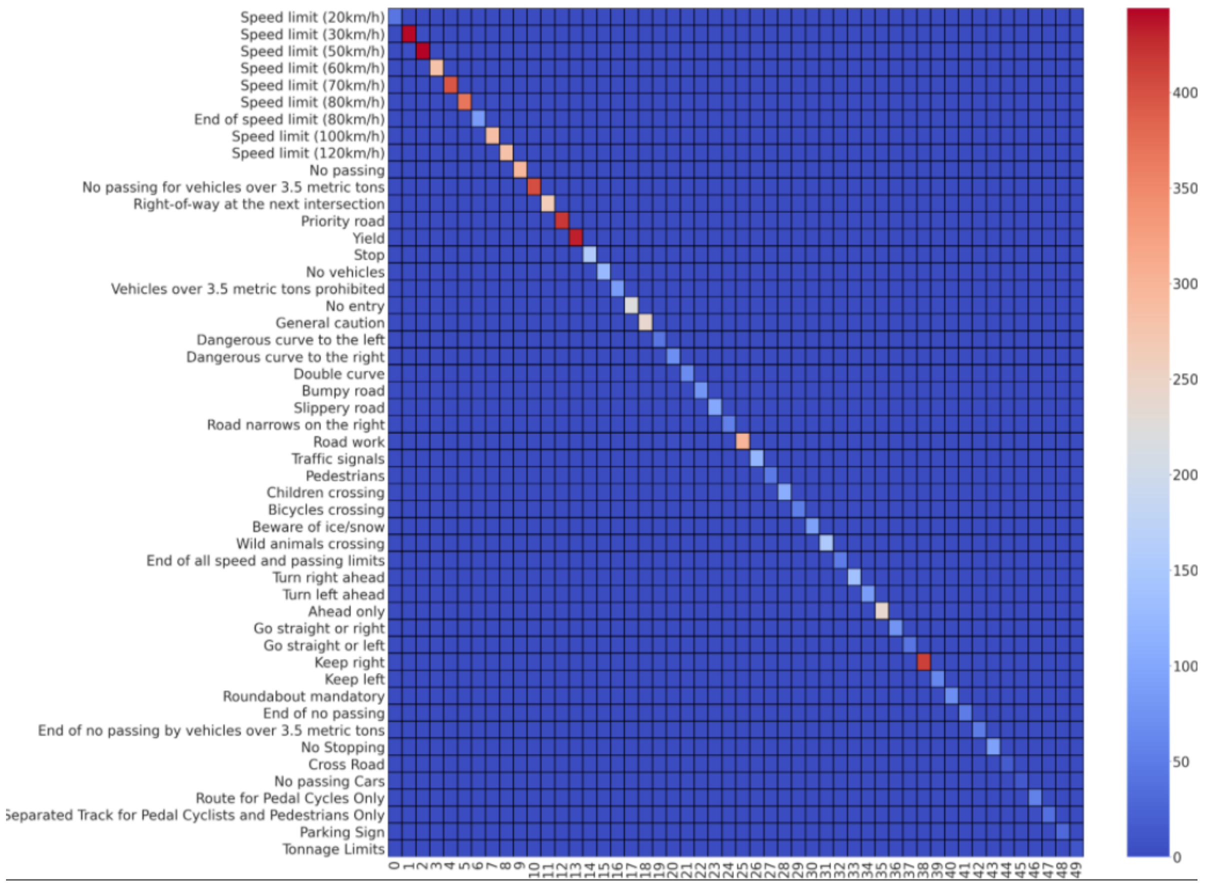


Post-experiment evaluation

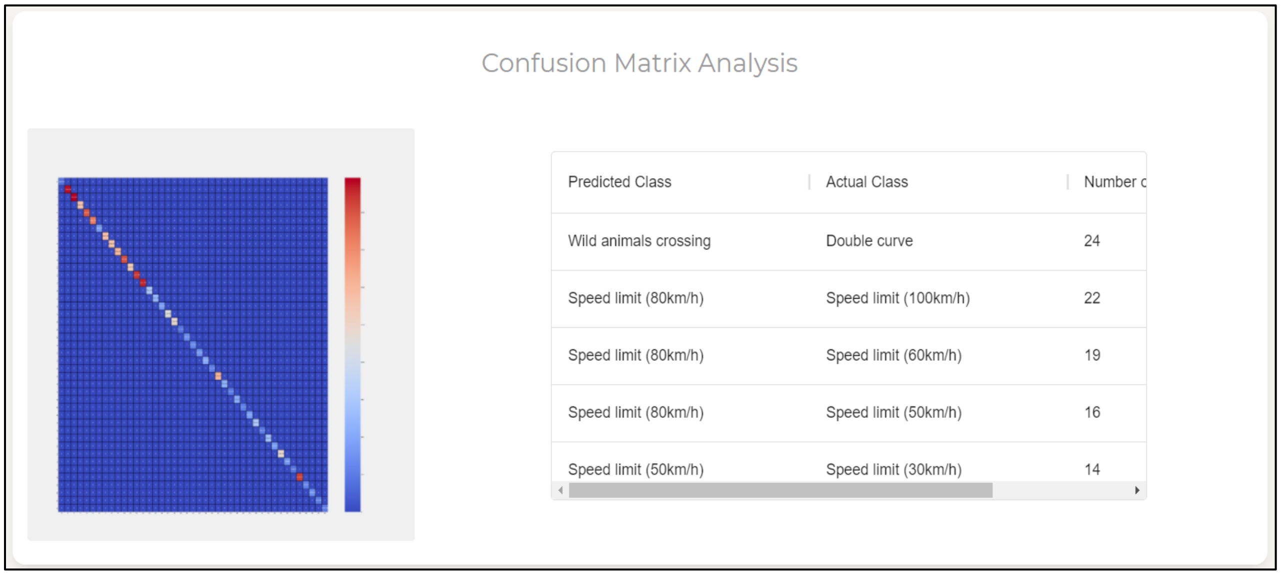
We provide a UI that enables the user to analyze and understand the pitfalls in the model and the dataset. We also suggest the user optimal settings for the next experiment which can improve the results. We provide the following features to the user for post-experiment evaluation:

Confusion matrix analysis

A Confusion matrix gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.



Confusion matrix displayed in UI


Confusion Matrix Analysis

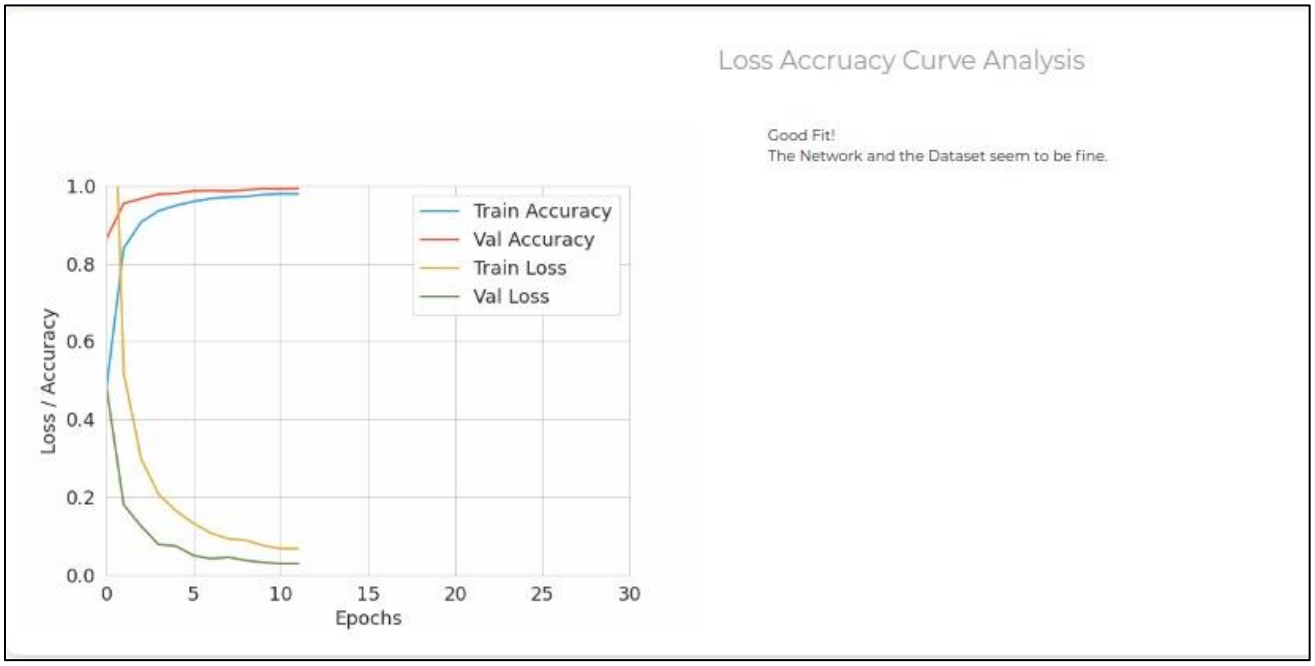
Predicted Class	Actual Class	Number c
Wild animals crossing	Double curve	24
Speed limit (80km/h)	Speed limit (100km/h)	22
Speed limit (80km/h)	Speed limit (60km/h)	19
Speed limit (80km/h)	Speed limit (50km/h)	16
Speed limit (50km/h)	Speed limit (30km/h)	14

UI for confusion matrix

Here the user can see the confusion matrix and the top 5 mis predicted classes with their actual classes and the number of such images.

Loss Accuracy Curve Analysis

We display the loss accuracy curve and analyze if the model is underfitting or overfitting. We then suggest possible steps that should be taken by the user to avoid underfitting or overfitting in the next experiment. The suggestions are based on changing the number of epochs, choosing a different optimizer, decreasing the dataset difficulty, altering the learning rate and depth of the network.



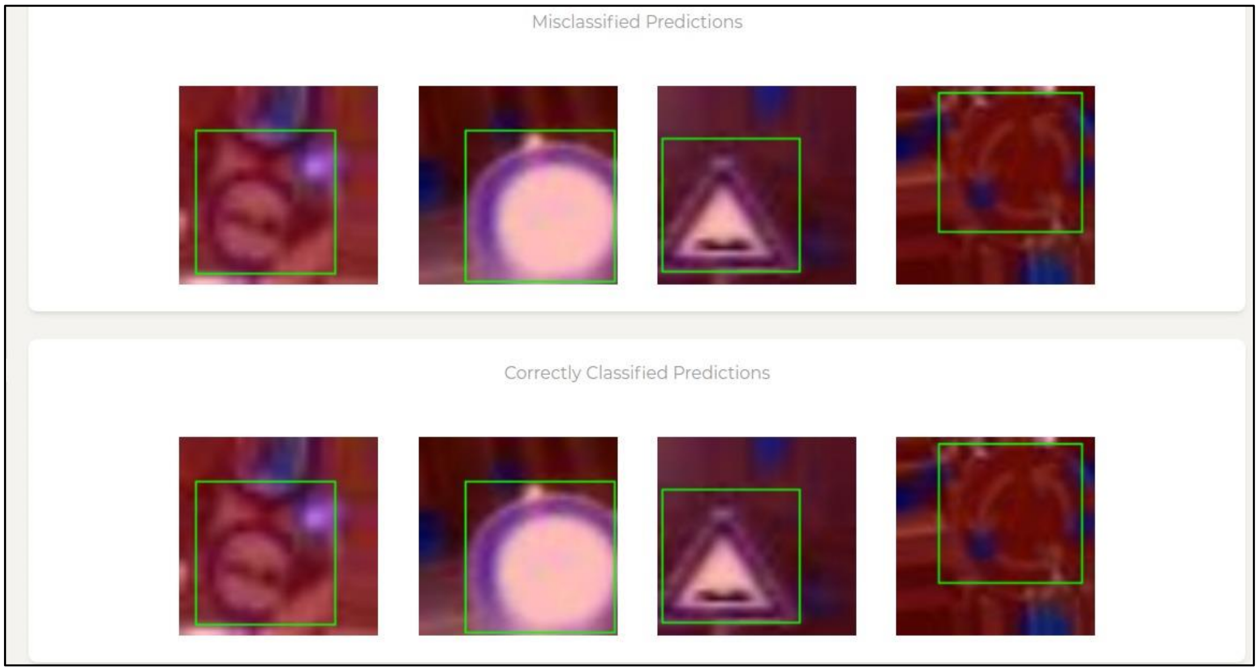
Loss Accruacy Curve Analysis

Good Fit!
The Network and the Dataset seem to be fine.

Explainable AI

We go a step ahead and give the user an option to use an AI that can explain the failures of the model. For this purpose, we use **Grad-CAM** (Gradient-weighted Class Activation Mapping).

Grad-CAM uses the gradients of any target concept, flowing into the final convolutional layer to produce a coarse localization map highlighting important regions in the image for predicting the concept.



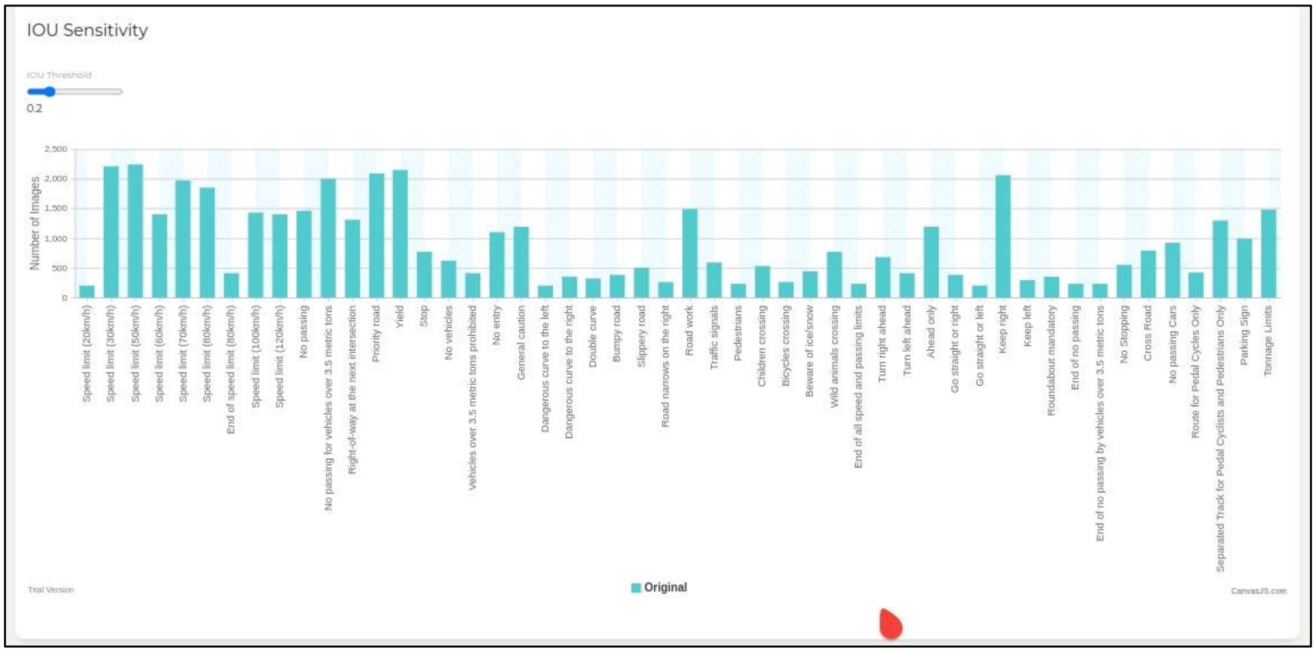
Methodology

We first find out the region of interest in our image using pre-trained models to create a bounding box. Then we use Grad-CAM to find out how much of the focus area lies inside the bounding box. We use IoU (Intersection over Union) to quantify this overlapping area.

Features

We display 4 Grad-CAM images for correctly classified images to see the important regions where the model is focusing. Similarly, we show 4 Grad-CAM images for the incorrectly classified images to see why the AI is failing to correctly classify the image.

We allow the user to select an IoU value range using a simple slider. The bar graph dynamically displays all the images class wise which have an IoU value in the selected range.


IOU Sensitivity
IOU Threshold
Number of Images
Speed limit (20km/h)
Speed limit (30km/h)
Speed limit (50km/h)
Speed limit (60km/h)
Speed limit (70km/h)
Speed limit (80km/h)
End of speed limit (80km/h)
Speed limit (100km/h)
Speed limit (120km/h)
No passing
No passing for vehicles over 3.5 metric tons
Right-of-way at the next intersection
Priority road
Yield
Stop
No vehicles
Vehicles over 3.5 metric tons prohibited
No entry
General caution
Dangerous curve to the left
Dangerous curve to the right
Double curve
Bumpy road
Slippery road
Road narrows on the right
Road work
Traffic signals
Pedestrians
Children crossing
Bicycles crossing
Beware of ice/snow
Wild animals crossing
End of all speed and passing limits
Turn right ahead
Turn left ahead
Ahead only
Go straight or right
Go straight or left
Keep right
Keep left
Roundabout mandatory
End of no passing
End of no passing by vehicles over 3.5 metric tons
No Stopping
Cross Road
No passing Cars
Route for Pedal Cycles Only
Separated Track for Pedal Cyclists and Pedestrians Only
Parking Sign
Tonnage Limits
Trial Version
Original
CanvasJS.com