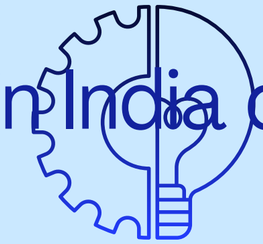
CAR ENHANCEMENT

Team: HackOverflow

PROBLEM



- According to an NCRB report more than 30% of people in India die due to **potholes**.

- Data on record with the Ministry of Road Transport reveals that **speed-breakers** in India cause 30 crashes every day.

- Road fatalities on account of weather conditions like **fog** run in thousand every year.Worst being that these accidents increase by 20 percent with every increasing year.

Goals

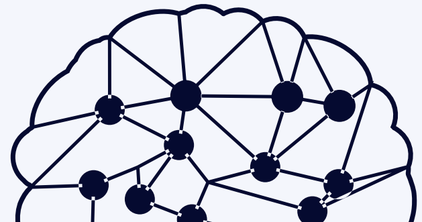
1. Obstacle avoidance on the road using AI-based Object Recognition Algorithm

2. Protection from high-beam lights coming from vehicles moving in front (in case of one - way lane) and those coming from the opposite direction (in case of two way lane)

Obstacle detection algorithm using AI based on Deep Learning Model of Convolution Neural Network (CNN) -

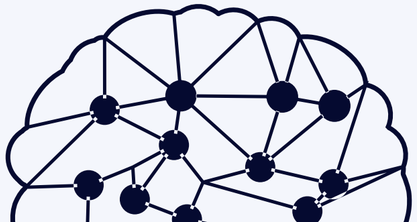
We are training our data model to detect objects in pictures that are similar to the ones in the data set that we have provided.

Creating class - specific heatmap off of a particular input image using Grad - CAM, a trained CNN, and a chosen class of interest.



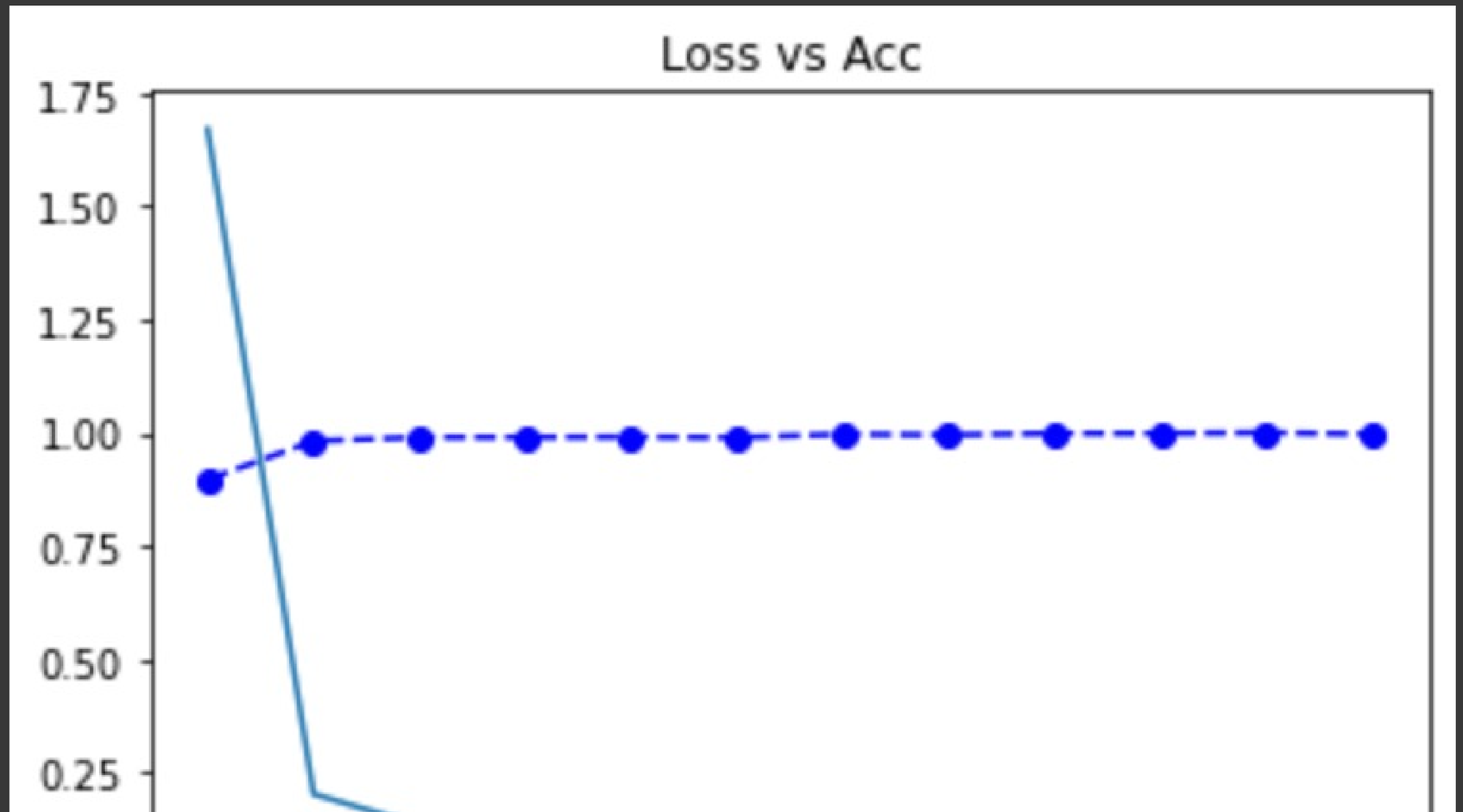
Currently we have 3 types of objects in our data set - potholes, speed breakers and fog - about 300 images of each set.

However, data set can be expanded to include animals such as dogs which are often a major reason for road casualties.



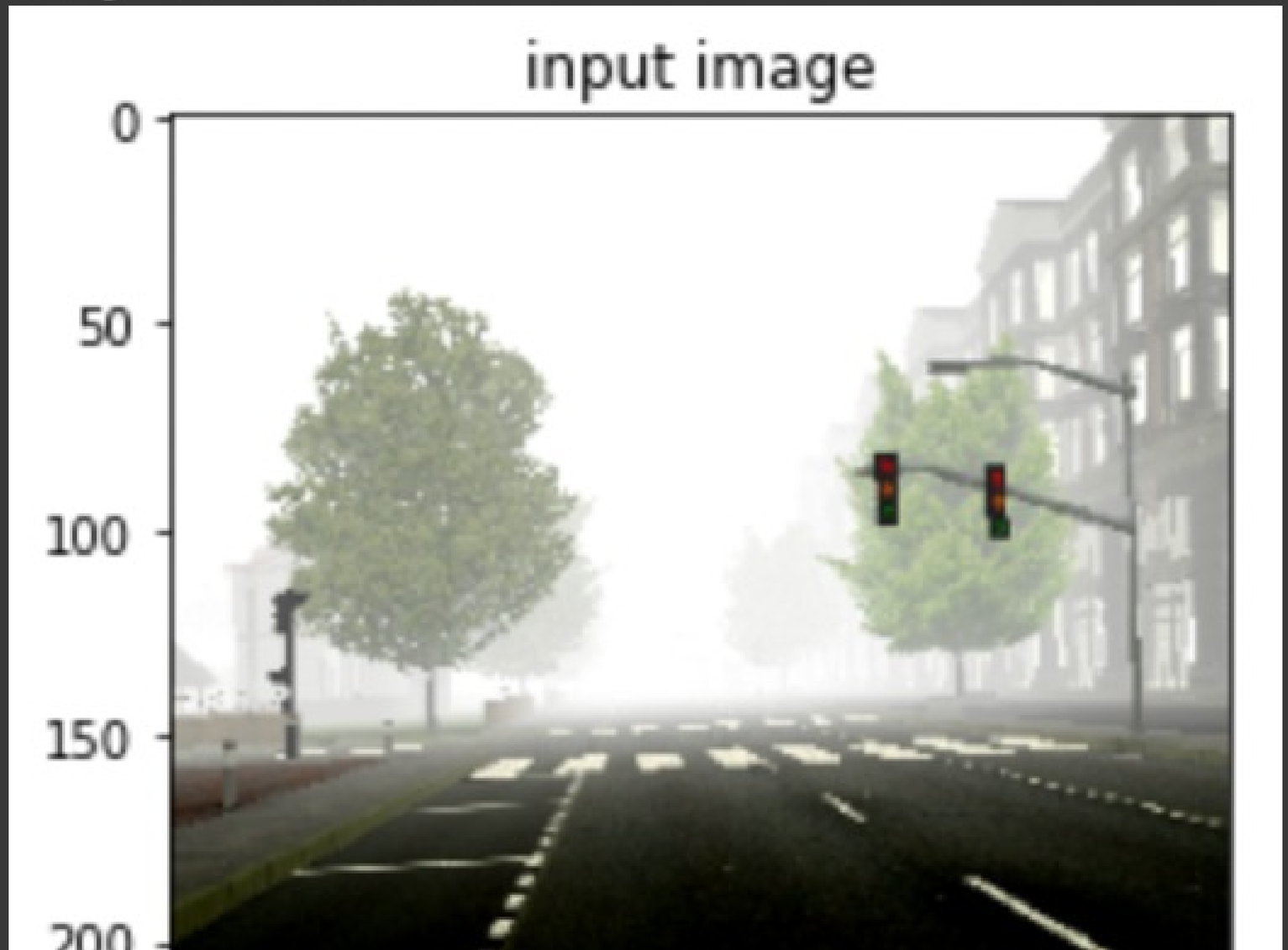
```
[ ]  plt.plot(h['loss'])
     plt.plot(h['accuracy'],'go--', c = "blue")

     plt.title("Loss vs Acc")
     plt.show()
```

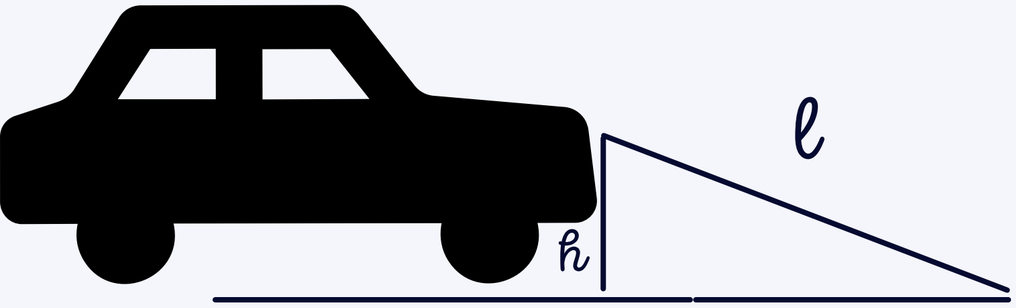


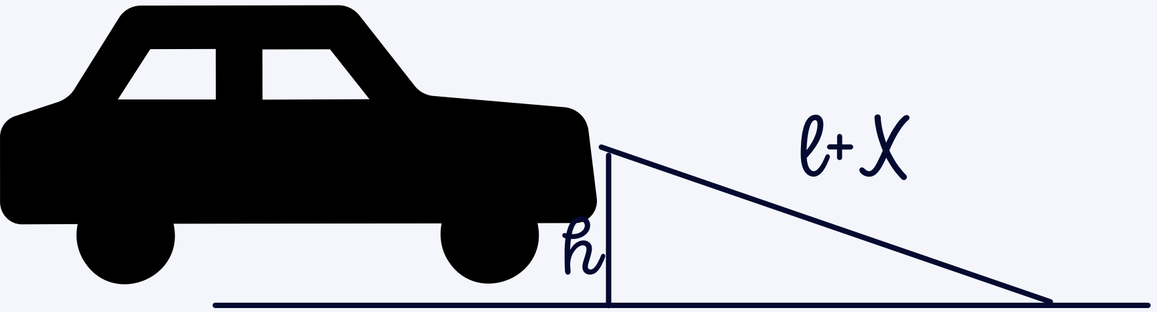
```
Epoch 4: accuracy did not improve from 0.98983
10/10 [==============================] - 14s 2s/step - loss: 0.0480 - accuracy: 0.9898
Epoch 5/30
10/10 [==============================] - ETA: 0s - loss: 0.0883 - accuracy: 0.9906
Epoch 5: accuracy improved from 0.98983 to 0.99063, saving model to ./best_model.h5
10/10 [==============================] - 16s 2s/step - loss: 0.0883 - accuracy: 0.9906
Epoch 6/30
10/10 [==============================] - ETA: 0s - loss: 0.0847 - accuracy: 0.9891
Epoch 6: accuracy did not improve from 0.99063
10/10 [==============================] - 15s 2s/step - loss: 0.0847 - accuracy: 0.9891
Epoch 7/30
10/10 [==============================] - ETA: 0s - loss: 0.0398 - accuracy: 0.9966
Epoch 7: accuracy improved from 0.99063 to 0.99661, saving model to ./best_model.h5
10/10 [==============================] - 15s 1s/step - loss: 0.0398 - accuracy: 0.9966
Epoch 8/30
10/10 [==============================] - ETA: 0s - loss: 0.0110 - accuracy: 0.9949
Epoch 8: accuracy did not improve from 0.99661
10/10 [==============================] - 14s 1s/step - loss: 0.0110 - accuracy: 0.9949
Epoch 9/30
10/10 [==============================] - ETA: 0s - loss: 0.0175 - accuracy: 0.9984
Epoch 9: accuracy improved from 0.99661 to 0.99844, saving model to ./best_model.h5
10/10 [==============================] - 16s 2s/step - loss: 0.0175 - accuracy: 0.9984
Epoch 10/30
10/10 [==============================] - ETA: 0s - loss: 0.0033 - accuracy: 0.9983
Epoch 10: accuracy did not improve from 0.99844
10/10 [==============================] - 14s 1s/step - loss: 0.0033 - accuracy: 0.9983
Epoch 11/30
10/10 [==============================] - ETA: 0s - loss: 4.8954e-04 - accuracy: 1.0000
Epoch 11: accuracy improved from 0.99844 to 1.00000, saving model to ./best_model.h5
10/10 [==============================] - 18s 2s/step - loss: 4.8954e-04 - accuracy: 1.0000
Epoch 12/30
10/10 [==============================] - ETA: 0s - loss: 0.0166 - accuracy: 0.9966
Epoch 12: accuracy did not improve from 1.00000
10/10 [==============================] - 16s 2s/step - loss: 0.0166 - accuracy: 0.9966
Epoch 12: early stopping
```

```
1/1 [==============================]
WARNING:matplotlib.image:Clipping inp
Fog Detected
```



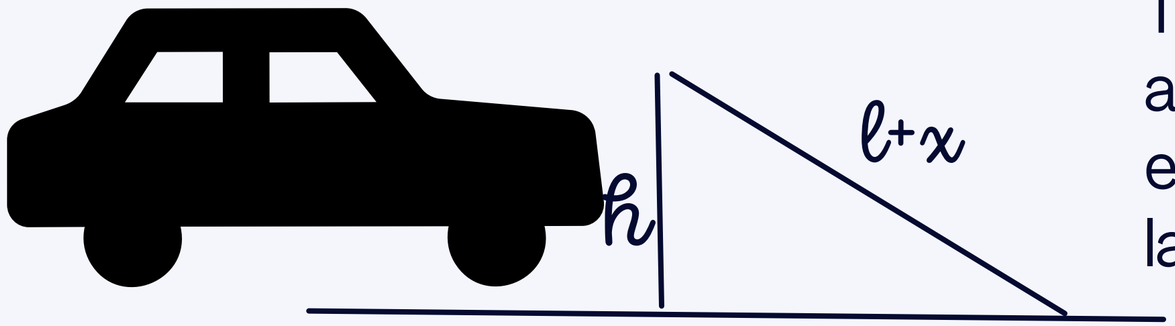
Working of ultrasonic sensor





A camera is placed at the front of the car. An Ultrasonic sensor placed with the camera calculates distance at a fixed angle theta, covering a range of angle.

As long as the distance remains a constant l (or very close to l), it indicates the road is clear.



When a pothole appears or the road is uneven, the slant distance is bound to increase by a certain value x, for a very small time interval,after which it goes back to the normal l.

The driver is hence alerted of the uneven road in advance, which can save the passengers from extensive jerks and from serious accidents at large.

APPLICATIONS

- A camera detects potholes and speed breakers and alerts the driver of the same .

- Identifies most road signs, and sends an alert to the driver every time one is spotted and identified.

- In case of Environmental conditions where the camera might not work properly , we use ultrasonic sensors to detect potholes and obstacles.

- In case of dense fog, can alert the driver of incoming patch of fog

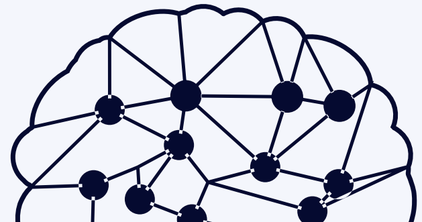
USING A LUX METER FOR PROTECTION FROM HIGH BEAMS

- A small lux meter is placed at a corner of the drivers headrest , facing the steering wheel .
- A safe limit for light intensity is input in the system.
- When the meter detects a light intensity greater than the safe threshold for a minimum of a pre-decided time interval , a small visor would drop down to protect the driver's eyes from the high beam light, and ensure proper vision to avoid accidents.

LIMITATIONS FACED

- Could not include more data sets such as road signs due to time constraints. However, code can be enhanced to include road signs that can have many applications.

- In case the car is traveling very fast we may not have enough time to identify the pothole and alert the driver, ie the car may pass the pothole in the time it takes for the model to detect the pothole , especially in situations where the outside environment is not ideal - for example , when it is raining



Thank You