ML Mini Project - Batch 2

18500139, Shali Suresh
185001170, Sruthi Bijoy
185001188, Vanathi G.
185001201, V Vrishin Vigneshwar

April 16, 2021

CAR LICENSE PLATE DETECTION

Problem Statement

To create and compare object detection models using ML and DL for identifying license plates in images of cars and plotting bounding boxes around them.

Introduction

Object Detection:

It is a type of visual recognition task where we have to not only recognize the object but also draw a bounding box around it. In short, our task is to identify and locate instances of an object in an image. For the chosen problem statement, we need to identify instances of license plates in the input image and draw a bounding box around the instance upon identification as shown in Fig. 1.

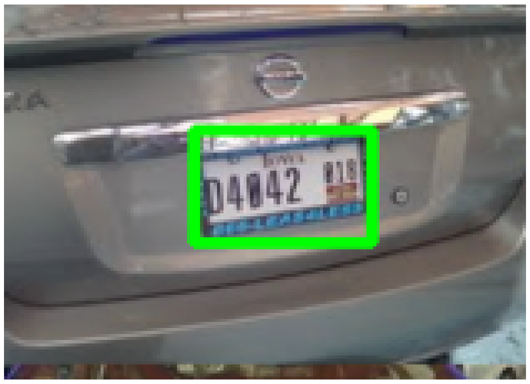


Figure 1. License plate identification

Methodology:

Machine Learning Model - SVM Classifier based on HOG Features:

We decided to implement the SVM classifier based on HOG because it is one of the best machine learning models for object detection. Histogram of Oriented Gradients (HOG) is a feature descriptor used in image processing, mainly for object detection. The principle behind the histogram of oriented gradients descriptor is that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. The x and y derivatives of an image (Gradients) are useful because the magnitude of gradients is large around edges and corners due to abrupt change in intensity and we know that edges and corners pack in a lot more information about object shape than flat regions. So, the histograms of directions of gradients are used as features in this descriptor.

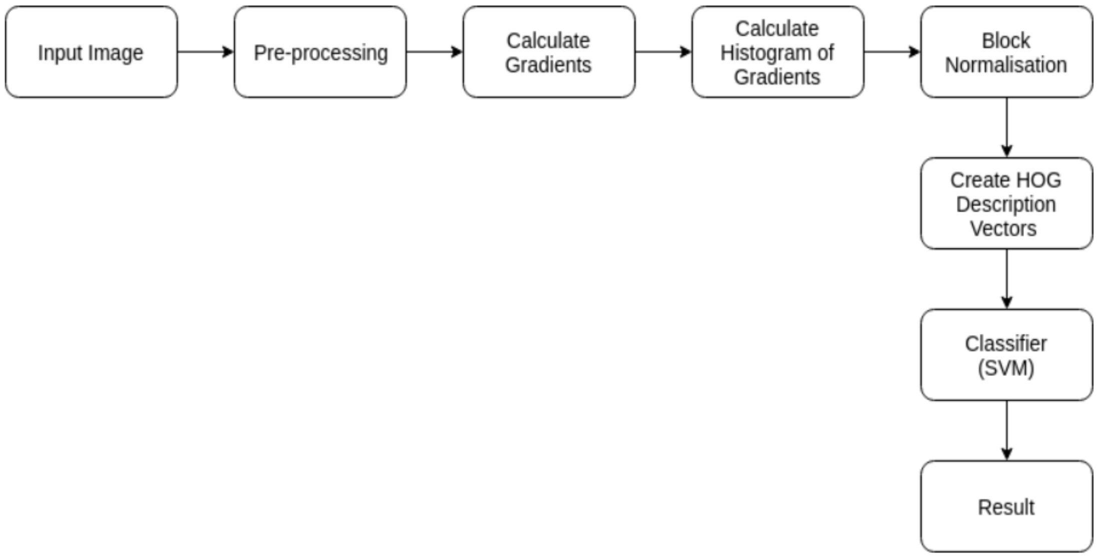


Figure 2. Flow of proposed SVM HOG model

In the proposed model, the data was preprocessed by performing the steps mentioned under the implementation section. The HOG for the preprocessed images were computed. These descriptors are the positive patches for object detection. Positive patches are images of license plates. For negative patches, images of camera, coins, clocks etc were converted into their corresponding HOG descriptors. Negative patches are images that don't have license plates in them. The SVM model was trained using these patches to identify license plates.

When input images are fed into the model, parts of the input image are checked for the presence of license plates using the sliding window algorithm. If the window contains a part of a license plate, a bounding box is drawn around the coordinates of the current window. The complete flow of the proposed SVM model is shown in Fig. 2.

Implementation:

1. **Data Preprocessing:**

   Step 1:

   The images are cropped using the bounding box coordinates to get the license plate patches from the images to be positive patches.

   Step 2:

   The cropped images are resized to 75x50.

   Step 3:

   The images are then converted into black and white.

   Step 4:

   Images of 'camera', 'text', 'coins', 'moon', 'page', 'clock', 'chelsea', 'coffee', 'immunohistochemistry', and 'hubble_deep_field' were used to extract negative patches.

   Step 5:

   The labels of the images of license plates are assigned a value 1 indicating that they are positive patches and the rest are labelled as 0.

   Step 6:

   The positive and negative patches and chained together.

2. **Tools Used:**
   a. Numpy
   b. pyplot
   c. cv2
   d. os
   e. glob
   f. etree

g. train_test_split
h. LinearSVC
i. GridSearchCV
j. hog
k. chain
l. data
m. transform
n. color
o. PatchExtractor

Deep Learning Model - CNN using VGG16:

We decided to implement the Convolution Neural Network (CNN) because we think it is one of the most efficient algorithms in this domain. CNNs show high performance in computer vision tasks including object detection. We have also used a pre-trained model, namely, VGG16, in our model. VGG16 is a pre-trained object recognition deep learning model network that uses a CNN to classify images.

The VGG16 internal architecture is shown in Fig. 3. It uses a CNN with 16 layers to recognize a 224x224 image. There are 2 convolution layers of 64 channels, followed by a maxpool layer, 2 convolution layers of 128 channels, another maxpool layer, and 3 convolution layers of 256 channels, and a maxpool layer. Further has 3 convolution layers of 512 channels, a maxpool layer, 3 convolution layers of 512 channels, and finally another maxpool layer. All the convolution layers have 3x3 kernels and the same padding. All the maxpool layers have 2x2 pool size and stride. The CNN is followed by 3 fully connected layers with ReLu as the activation function.

The output of the VGG16 model is flattened to a 18432x1 feature vector. The flatten layer is followed by 4 dense layers with output sizes 128x1, 128x1, 64x1, and 4x1 with activation functions as relu for the first 3 layers and sigmoid function for the last dense layer.

The model takes an image as input and identifies the position of the license plate in the input image if any, and predicts the coordinates of the bounding box of the identified license plate. It predicts the bottom-left and top-right coordinates of the bounding box rectangle. The dimensions of the layer dense layers 4x1 because the model predicts these 4 values as its output. The complete architecture diagram of the proposed model is shown in Fig. 4.

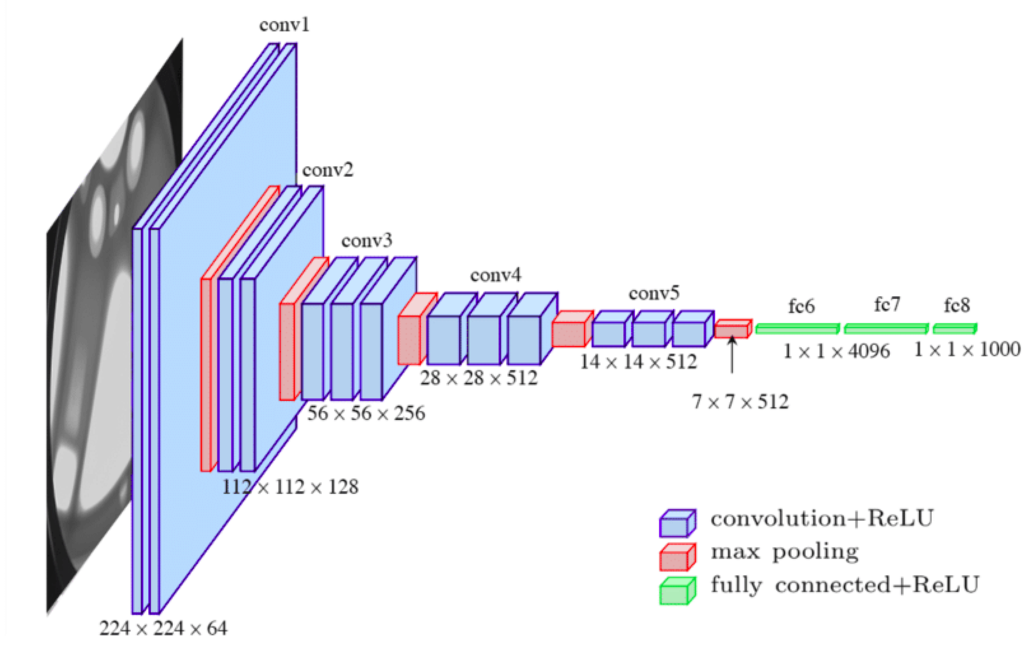


Figure 3. VGG16 architecture diagram.

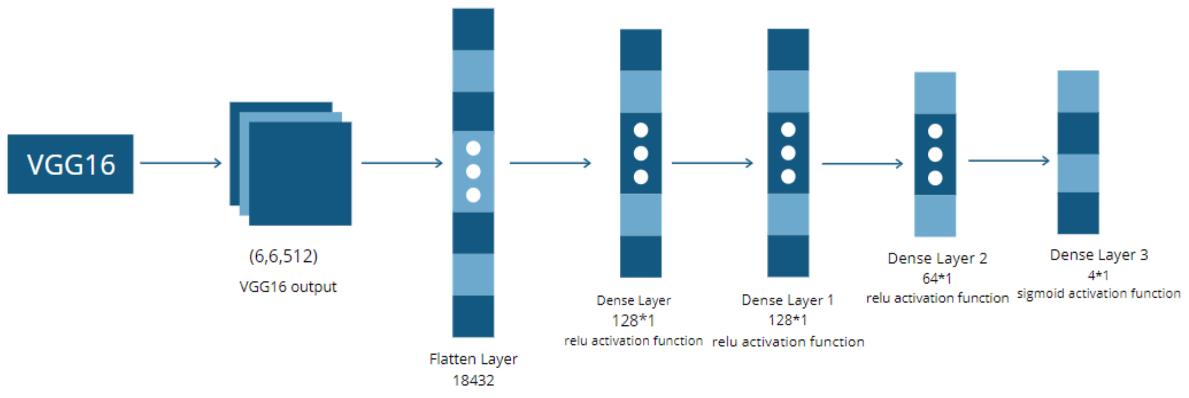


Figure 4. Architecture diagram of the proposed model.

Implementation:

1. Data Preprocessing:

Step 1:

All the images in the dataset are resized to 200x200x3.

Step 2:

The coordinates of the bounding boxes are resized according to the factor by which their corresponding images were resized in Step 1.

Step 3:

The images are converted into numpy arrays before training.

Step 4:

The values of the numpy arrays and bounding box coordinates are normalized.

2. Tools Used:

a. pandas
b. numpy
c. pyplot
d. seaborn
e. cv2
f. os
g. glob
h. etree
i. train_test_split
j. Sequential
k. Dense
l. Flatten
m. VGG16

Performance Metric:

Intersection over Union (IoU):

Intersection over Union is an evaluation metric used to measure the accuracy. Any algorithm that provides predicted bounding boxes as output can be evaluated using IoU.
Intersection over Union (IoU) is a metric that allows us to evaluate how similar our predicted bounding box is to the ground truth bounding box. The idea is that we want to compare the ratio of the area where the two boxes overlap to the total combined area of the two boxes. Fig. 5 shows an example of detecting license plates in an image. The predicted bounding box is drawn in red while the ground-truth bounding box is drawn in green. IoU is used to compute the Intersection over Union between these bounding boxes.

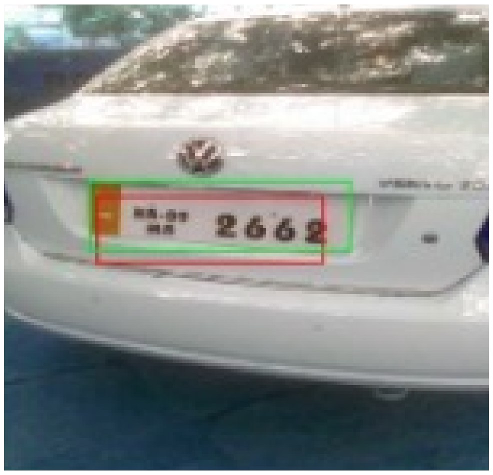
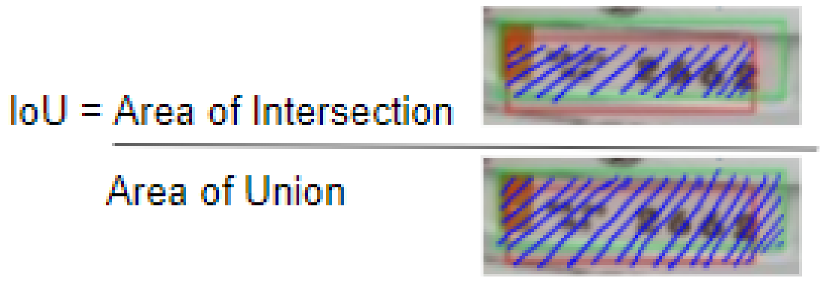


Figure 5. Predicted bonding box (red) and ground truth bounding box (green).

Unlike regular classification problems that predict class labels, object detection predicts the bounding box coordinates. We cannot simply determine the performance of an object detection model by checking whether the output is 'correct' or 'incorrect' as in the case of classification problems.
It is extremely unlikely that the coordinates of the predicted bounding boxes exactly match the coordinates of the ground truth bounding boxes.

So, we use IoU to evaluate our model based on the extent of overlapping between the two bounding boxes. Heavy overlapping corresponds to a better model.

IoU was calculated using the following formula:

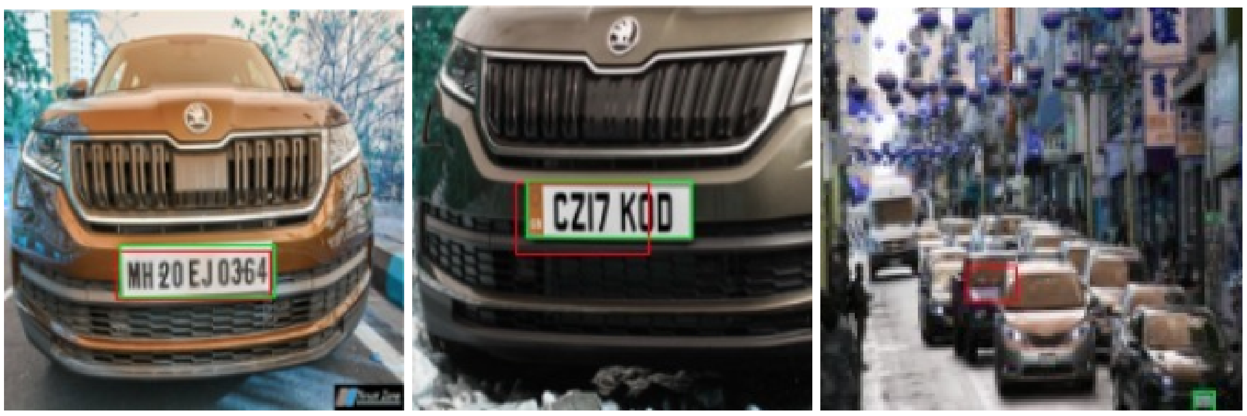


Results and Comparison:

1. Predicted bounding boxes:

- DL Model:

  Fig. 6 shows the predicted bounding boxes for the deep learning model using VGG16. While most of the boxes we predicted correctly (Fig. 6.a and 6.b), it can be inferred that the VGG16 model does not perform well for images that have more than one license plate instance in it (Fig. 6.c).



- ML Model:

  Fig. 7 shows the predicted bounding boxes for the machine learning model using HOG and SVM. The performance is not as accurate when it comes to identifying "where" the boxes are inside an image with a box, but is very very accurate when it comes to answering the question "whether the image has a license plate or not", i.e. plain classification is really accurate.

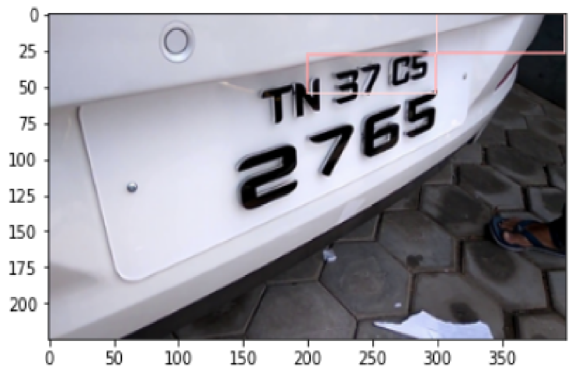
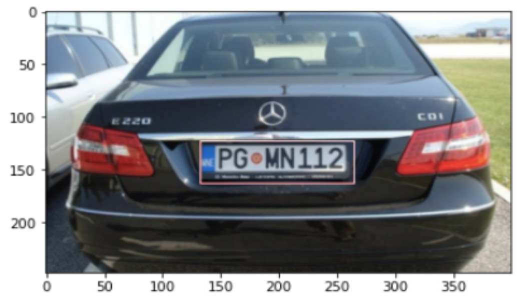
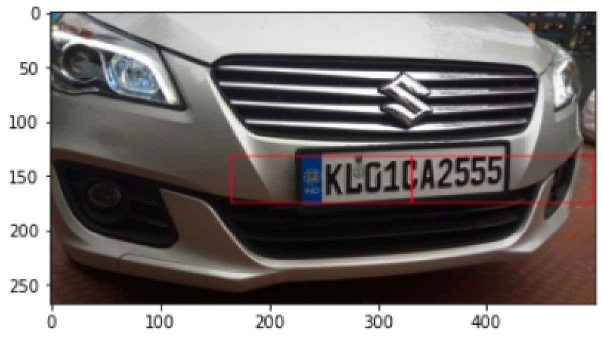




Figure 7. Predicted results of the proposed ML model.

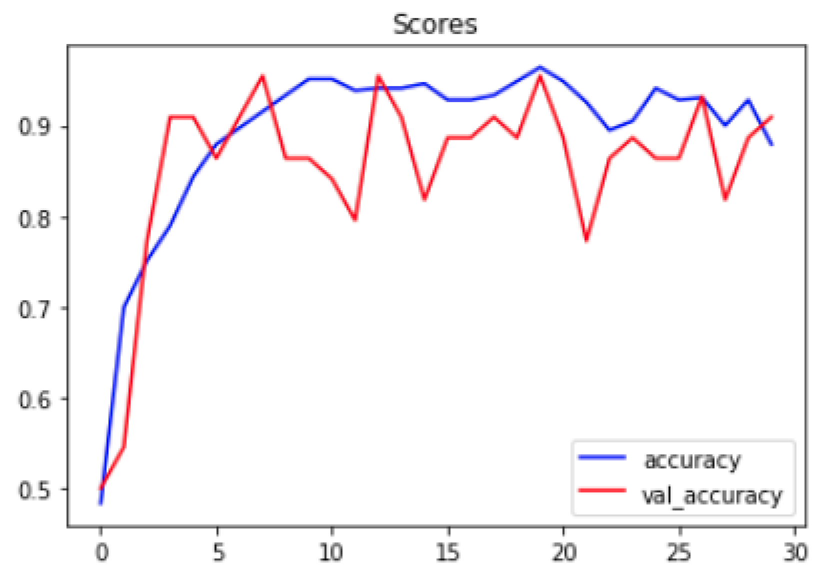
2. Accuracy:

Table 1 shows the comparison between the 2 proposed models based on accuracy. Though the ML model gave better accuracy values than the DL model, upon further studying, it was understood that while the accuracy of the DL model is based on how accurately the model was able to predict the bounding box coordinates, the accuracy of the ML model was based on whether the model classified an image as having license plate instances or not. It was also observed that though the ML model correctly classified some images as positive, what the model identified as license plates were not really license plates. It basically falsely identified other objects as license plate instances as shown in Fig. 7.

ML Model	DL Model
98%	90.2%

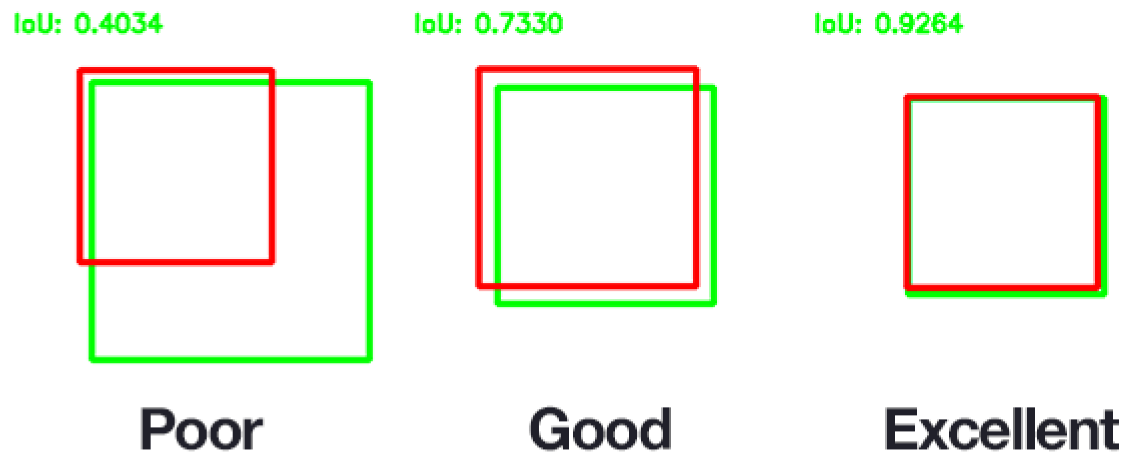
Table 1. Accuracy of the proposed models.

Since the criterion for measuring accuracy is different for both the models, it is not a suitable metric for comparison, but gives one a reasonable estimate.



A visual comparison between our val accuracy scores and accuracy scores obtained for the VGG16 model is shown by the above graph. As expected, for the most part, accuracy is well above val accuracy. We can also see that the scores converge to become almost the same with very little residue as the number of epochs increases. This tells us that #epochs = 30 is about a good time we stop training our model, so that our model does not start overfitting and remains fairly stable.

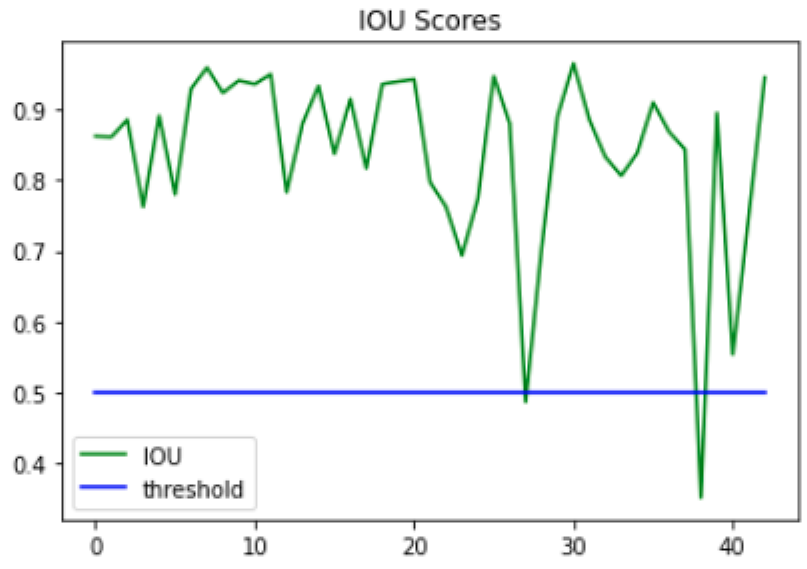
3. Intersection over Union:



As previously mentioned, **Intersection over Union** is an evaluation metric used to measure the accuracy of an object detector on a particular dataset. The above image depicts the obvious relation that a good fit implies a high score closer to 1, and vice versa otherwise.

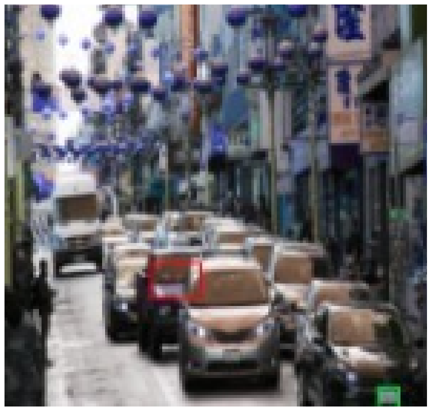
<u>In the DL paradigm, a IOU score of over >= 0.5 on average suggests that the model has been performing well</u> in general which was our objective when we built the model. This is also reflected in our findings below:

Our deep Learning model's IOU scores,



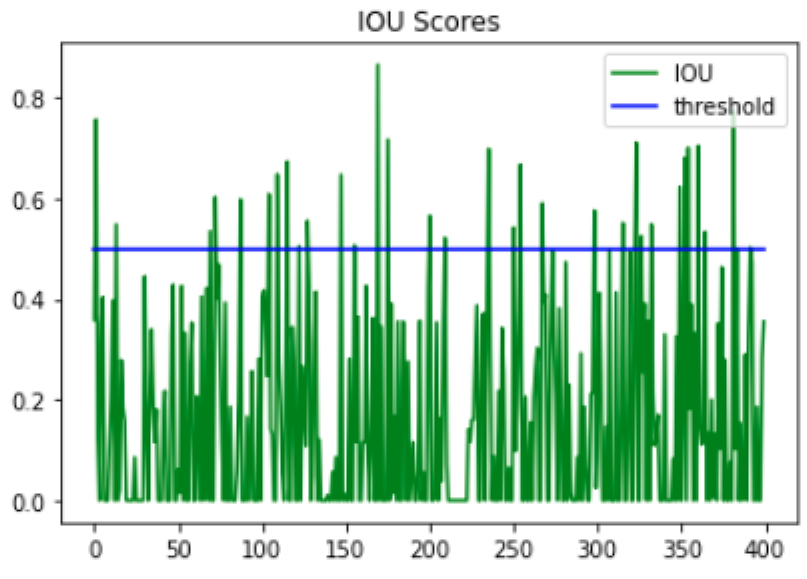
As we can see the model seems to have got just 1 or 2 score below 0.5 which can be neglected as being extreme data points. <u>On the brighter side, our model boasts a very high average of 0.8375846286</u>

Spoiler: guess which image resulted in a 0.33 IOU score? :)



Did you guess this? If yes, pat yourself, because we did pat ourselves! As explained above, we exactly predicted this outcome because our VGG16 was expected to fail when the image had a lot of grain and noise in it. And not so surprisingly it did fail with a bad IOU score of 0.33. Well, getting a good score even in such a bad image is one task for the future, we have decided to let this one slide for the moment.

Meanwhile our Machine learning model, provided equally interesting results when it came to the IOU scores.



Surprising isn't it? Well it wasnt so surprising for us.

Remember how we made a big issue of a single 0.33 IOU score from a DL mode. It was done with a reason. A reason to show these absurd Ml model IOU scores that our model could come up with while solving this problem.

As we can see, the ML model had certain inherent issues, for instance

1. The no of negative patches when compares to the positive patched was extremely high
2. There was no quality negative patches that could be extracted from several combination of patches that we fed to the model, which led to lesser accuracy
3. The box merge NMS algorithm that we used did a decent job at best and the sliding window parameter used towards the end didn't compliment to the rest of the model really well, even though these were some of the best tools available in the market

4. Conclusion:

What can be derived from the analysis of using DL and ML to try and solving the same problem is the following:

1. For an object detection problem DL proved to be extremely superior to ML.
2. The main advantage of a CNN was its ability to learn complex features
3. Even one of the best HOG SVM object detection models that ML boasts, can at most just classify an image if it contains the desired object or not. However when it came to actually checking if it had predicted the bounding box well, it was clear as daylight that it didn't do that great of a job as compared to CNN.(refer IOU scores for the same)

4. The ML model moreover requires a lot of preprocessing to assist in its prediction and training, however the CNN DL model requires very little preprocessing
5. The residues and the errors from the ML model were significantly large as compared to the DL model, which was dissatisfying.