INDUSTRIAL TRAINING REPORT

Illegal Parking Detection

Submitted in partial fulfillment of the

Requirements for the award of

Degree of Bachelor of Technology in Computer Science & Engineering

Submitted By

Name: **Jaskaran Singh**

University Roll No.: **02814802717**

Submitted To:

Department of Computer Science & Engineering

MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY

GGSIPU, DELHI

ACKNOWLEDGEMENT

It is my pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced my thinking, behaviour and acts during the work. First and foremost, I express my sentiment of gratitude to **Mr Ratnesh Bhardwaj,** Machine Learning Developer at Mekvahan who has been a continuous source of inspiration as my intern mentor. Without his constant guidance and suggestions, this report would have been nowhere near completion. My gratitude for his trust and generosity goes beyond words.I am highly indebted to **Mr Sanjeev Kumar** who took confidence in me and provided me with the opportunity to work as an Intern at **Mekvahan .** I had a wonderful and an unforgettable experience being part of such a lovely and lively team. I express my sincere gratitude to Computer Science Department, M.A.I.T worthy Principal for providing me an opportunity to undergo summer internship at Mekvahan . Lastly, I would like to thank the almighty and my parents for their moral support and my friends with whom i shared my experience from the internship and received lots of suggestions that improved my quality of work.

Thanking you

Jaskaran Singh

ABSTRACT

This report is prepared to fulfil the requirement of the Bachelor of Technology program of Maharaja Agrasen Institute Of Technology on Illegal Parking Detection at MekVahan Pvt Ltd.

In these 4 weeks of internship I learned how to work on a big project using Python (Tensorflow , Keras , Numpy) for Deep Learning. I learned how to deploy my model as an API on the local system. I learned ways to make our project more secure, work as a team and also how to maintain a codebase on open source like GitHub. It was a learning experience throughout my internship with MekVahan Pvt Ltd. The mentors were very helpful in resolving any doubt and encouraging pushing our limits.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

OBJECTIVE

With the high percentage of vehicle ownership in the India, parking has become a conflicting and confusing situation for the people.

Cars are parked everywhere in a reckless manner around us which has become the root cause of all the major traffic jams.

With the growing number of vehicles, this problem of illegal parking is increasing day by day.

Even if the cars are parked illegally there are no measurements taken in order to curb it. Not much people are fined and therefore keeps on parking illegally.

People don't know about the rules that whether to park a car at a particular location or not and in turn park anywhere they like causing traffic jams and even accidents in some cases.

CURRENT SYSTEM

Current solution that is present to illegal parking detection is very limited and require videos or live videos of the surveillance then this video looks for the hardcoded places in the image/frame to look if a vehicle is currently present in that place or not and if it is then it looks for the time it remains there .After a certain period of time it gives the illegally parked detection sign to the motor. The detected place is then send to the official which checks the video to confirm if the vehicle is parked illegally or not.

REQUIREMENT OF AMENDMENTS

We want to make a robust self decision making program that is not hardcoded for the places to look upon for the illegally parked vehicles and most importantly we need to identify the parking status using an image to make it robust and easy manageable .

Hence we need to make an AI based approach to solve this problem of classification , hence a model need to be integrated with the website to serve the purpose which should be able to inference the parking status of vehicle by just processing the image.

SOLUTION

Using AI based approach to detect illegal parking of vehicles (Cars) from an image.

Our model will receive an image of parked car through the user of the app , then our model will try to predict the status of parking based on the background details (Traffic Signs , Crowded (market) area, Open field, Traffic , On road , number of side cars etc.).

Our model will do the prediction for each car that is present in the image.

DATA COLLECTION

TYPE OF DATA

Deciding the data is the most important part of the whole project . It is not just some pictures of the Illegally parked and legally parked vehicles but more than that . In India there are no specific rules , like there may not be any “NO-PARKING” Sign , but it still be called illegally parked because of the fact that it is causing trouble to the near by area. hence there are no hard rules to follow to classify the parked vehicle as legal or illegal .

Hence the parking problem can also be viewed as if the parked vehicle is causing any trouble to the near by area or not , hence keeping this in mind to classify the parking status we are going to look at the surroundings of the parked vehicle , hence we need the images of the parked vehicles with both the status and with as much background as possible to clearly judge if the parked vehicle is causing any trouble or not .

TOTAL DATA

A total of 686 images were collected in which 335 were of legally parked cars and 351 were that of illegally parked .

Out of these 686 images 200 were collected from Connaught place and rest were collected from Shastri park area .

Now randomly 25% of the images were selected ,that is 269 images for the testing phase .

Now the training data is 417 images , which is way too low for training .

Some of the images consists of more than one vehicle in the image hence we can separately predict for each vehicle in that way it expands our data . After this step our training data becomes 1047 images .

Now image augmentation can be applied like zoom in , rotation ,flip . In this project 5 such processes are applied making it to 5235 images .

DATA LABELLING

Each image is labelled manually with the help of the team . First SSD is run on each image then it gives the coordinates of the detected image then the function asks for the label .

Single image may contain multiple cars hence label for each car is to be given .A pickle file is generated which contains the data in the format of a list with each list item be a tuple consisting of the image name , number of cars in the image , coordinates of the cars,labels of the cars .

SNAPSHOTS OF DATA


IND DL7C L 5736
DL 1CW 8836


S P CHOPRA & Co
CHARTERED ACCOUNTANTS


DL 9C W 6748
UP 16 Y 9693

TOOLS AND TECHNOLOGY USED

The application design is an integrated entity comprising of mainly the front-end (client side), back-end (server side) and database.

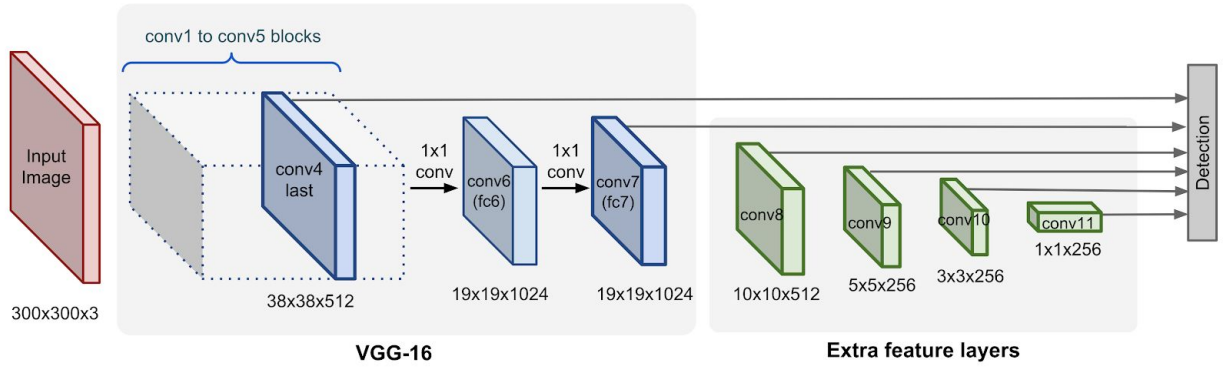
Open CV (Open Source Computer Vision)

- Open CV is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel).
- In this project Open cv is used for image augmentation task and image pre-processing .
- Image augmentation parameters that are generally used to increase the data sample count are zoom, flip, rotation.
- Image pre-processing involves normalizing the pixel value , blacking out the detected car using SSD model .

SSD Model

- **SSD** is a single-shot detector. It has no delegated region proposal network and predicts the boundary boxes and the classes directly from feature maps in one single pass. To improve accuracy, **SSD**introduces: small convolutional filters to predict object classes and offsets to default boundary boxes

- Model Architecture :

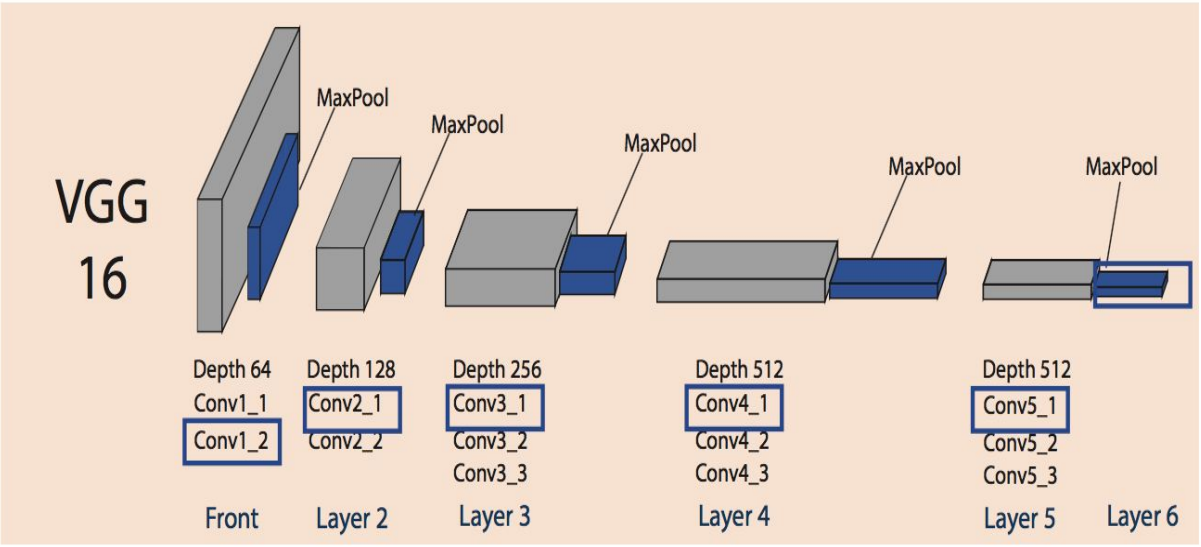


Here Pre-trained SSD Model is used to detect the cars in the image and gives the bounding boxes of the cars .

Predicting Model

- It consists of a feature extractor that is a pretrained model VGG16 , which is trained on image-net dataset (consist of million of images).
- It consist of 16 layers , but here we did not included the fully connected layers to extract features.
- The last layer is set to trainable.
- The extracted features and then passed through a Dense model or Linear SVC model.
- The Dense model consist of 5 layers or the Linear SVC is a Support Vector Classifier.

VGG16 Architecture :



jupyter model.py 07/31/2019

File Edit View Language

```python
import keras as ks

def create_model(input_shape):
    base_model=ks.applications.VGG16(include_top=False,input_shape=input_shape)

    trainable=False
    for layer in base_model.layers:
        if layer.name == "block5_conv1":
            trainable=True
        layer.trainable=trainable

    model=ks.models.Sequential()
    model.add(base_model)
    model.add(ks.layers.Flatten())
    model.add(ks.layers.Dropout(rate=0.25))
    model.add(ks.layers.Dense(1024,activation="relu"))
    model.add(ks.layers.BatchNormalization())
    model.add(ks.layers.Dense(512,activation="relu"))
    model.add(ks.layers.BatchNormalization())
    model.add(ks.layers.Dense(128,activation="relu"))
    model.add(ks.layers.BatchNormalization())
    model.add(ks.layers.Dropout(rate=0.25))
    model.add(ks.layers.Dense(64,activation="relu"))
    model.add(ks.layers.BatchNormalization())
    model.add(ks.layers.Dense(1,activation="sigmoid"))
    opti=ks.optimizers.adamax(lr=0.0001)
    model.compile(optimizer=opti,loss=ks.losses.binary_crossentropy,metrics=["accuracy"])

    return model
```

TECHNICAL CONTENTS

WORKING

The project is currently being integrated with the website .

USER SIDE

Our model will classify the cars from the images clicked by the users into two different categories that is, legally and illegally parked cars using deep learning.

The user need to upload the image of the vehicle from a certain distance to get the more of the background which helps the model in delivering good results.

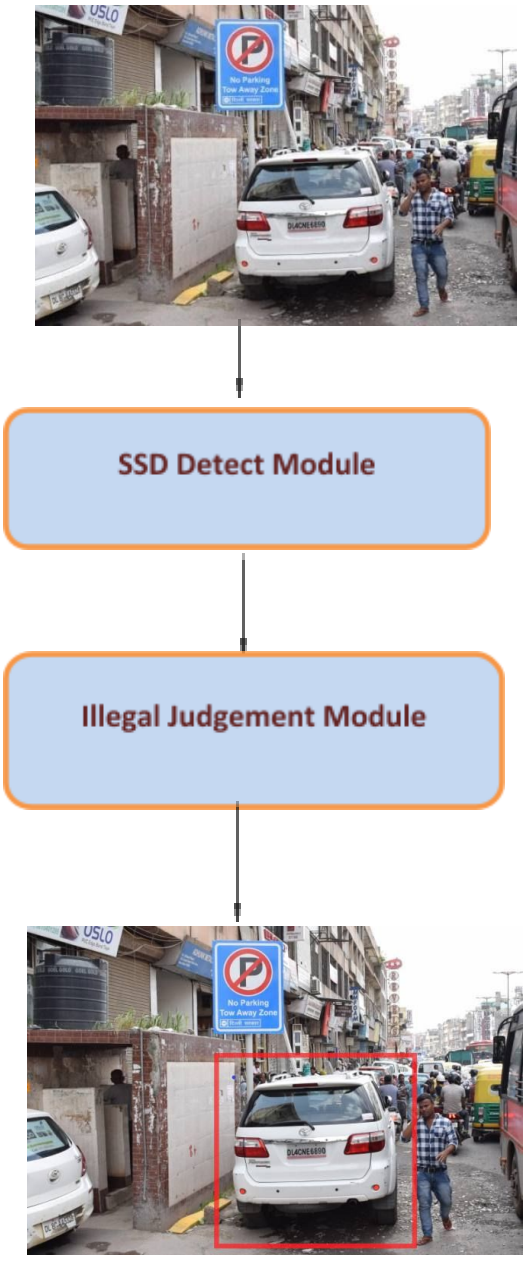
To confirm that the vehicle is at rest a video os 2-3 second may be uploaded.

EXECUTIVE SIDE

All the illegally parked predictions will be send to an executive to approve the e-challan ,

which can be done by the number plate of the vehicle which will be detected.

EXPLANATION



Steps Wise Explanation of our System

- The input of our system is a image captured by the phone App.
- The image captured in the above step is fed into our Model.
- Now, the Model have two tasks to do:
    - Object Detection
    - Object Classification

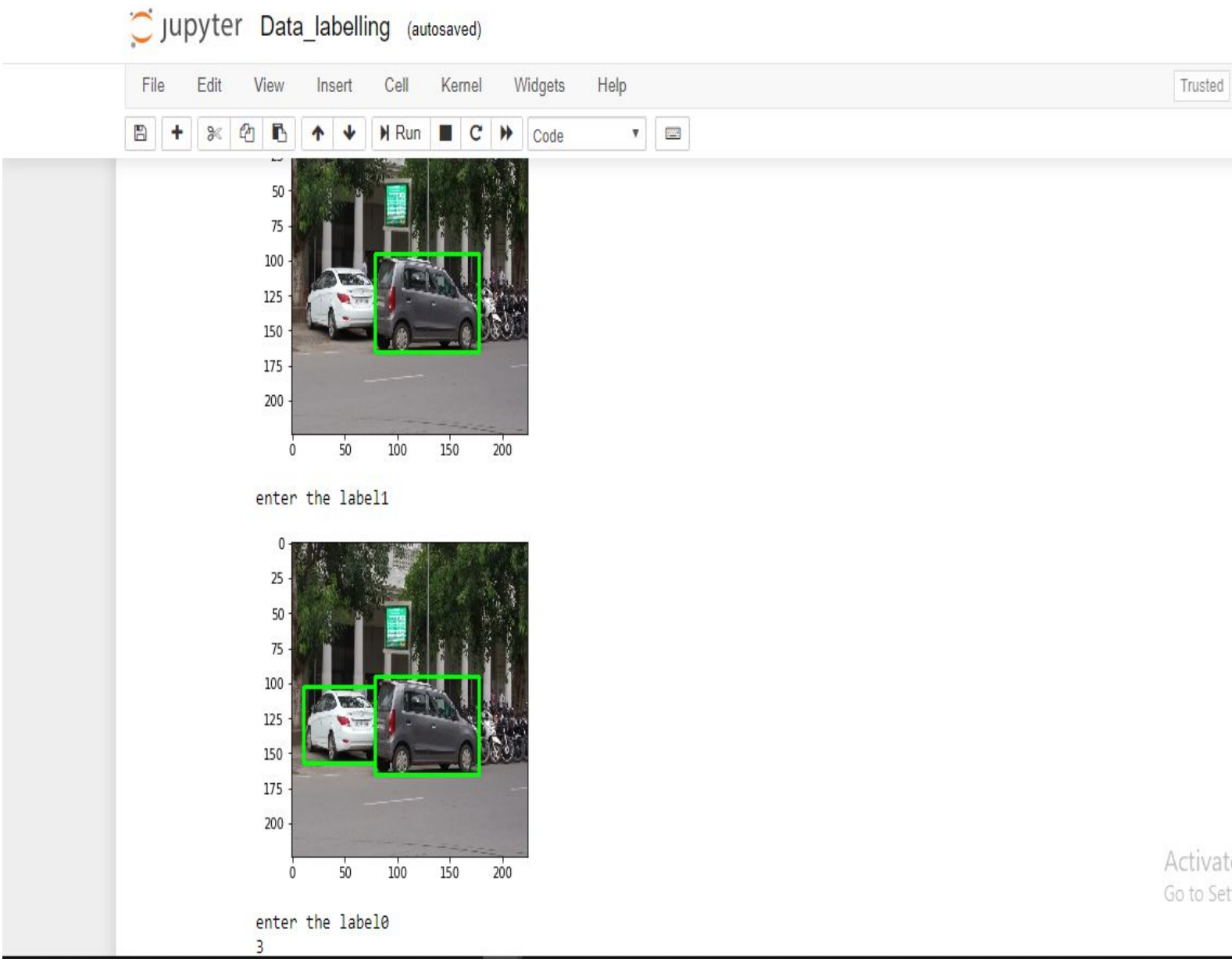
Object Detection : In this step the model is going to detect the location of the object/objects(car/bike).

Object Classification : Now we need to classify if the vehicle is parked illegally or not.

The model is designed to take the following things into consideration :

1. Background Details like open ground, road side area , parkside etc
2. Crowdedness
3. Detect for any traffic signs

SNAPSHOTS



Data Labelling

Data Augmentation ***(original)***

Augmented (crop and zoom , rotated (7 deg), rotated (-7 deg), scaling (1.2),flip)

Training Phase

jupyter train_colab (autosaved)

Logout

Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run Code

In [0]:
```
model=create_model((224,224,3))

batch_size=128
epochs=5
steps_per_epoch=int(len(train_lab)/(batch_size))
```

In [35]:
```
model.fit_generator(preprocess(train_lab,"./images/",batch_size=batch_size),steps_per_epoch=steps_per_epoch,epochs=epochs)
```

```
Epoch 1/5
40/40 [==============================] - 161s 4s/step - loss: 0.6273 - acc: 0.6684
Epoch 2/5
40/40 [==============================] - 160s 4s/step - loss: 0.4257 - acc: 0.8035
Epoch 3/5
40/40 [==============================] - 159s 4s/step - loss: 0.3313 - acc: 0.8574
Epoch 4/5
40/40 [==============================] - 162s 4s/step - loss: 0.2569 - acc: 0.9000
Epoch 5/5
40/40 [==============================] - 166s 4s/step - loss: 0.2015 - acc: 0.9279
```

Out[35]: <keras.callbacks.History at 0x7f7cd25b7fd0>

In [21]:
```
gen=preprocess(test_lab,"./images/",batch_size=len(test_lab))
test_data,test_labels=next(gen)

preds=model.predict_classes(test_data)
print(classification_report(test_labels,preds))
```

	precision	recall	f1-score	support
0	0.79	0.87	0.83	103
1	0.80	0.69	0.74	77

Testing

jupyter testing (autosaved)

File Edit View Insert Cell Kernel Widgets Help

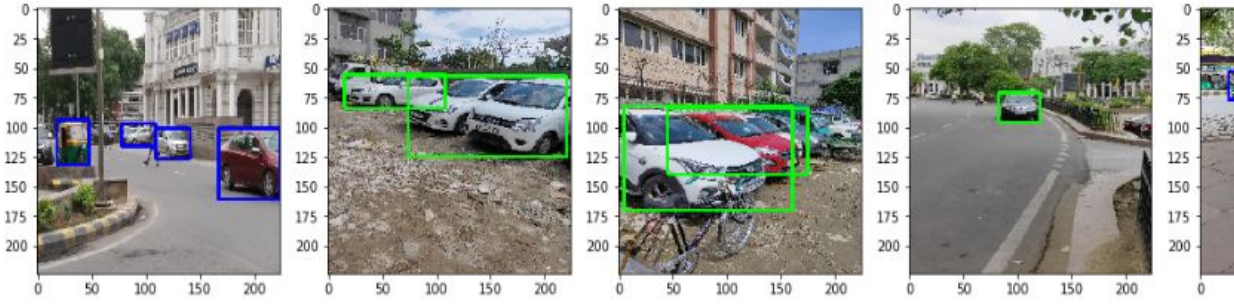
Run Code

In [64]:

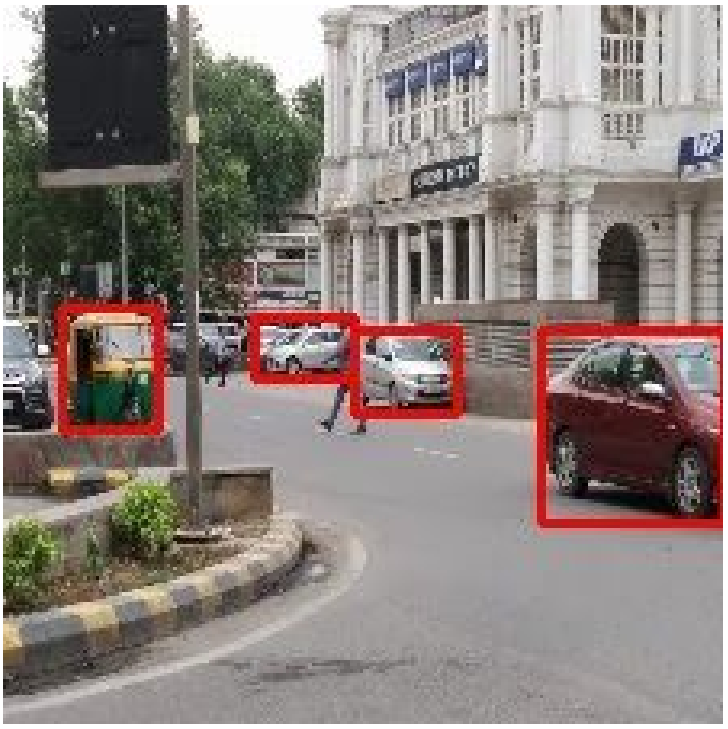
```
fig=plt.figure(figsize=(20,20))
num=1
np.random.shuffle(test)
for test_img in test[:5]:
    ax=fig.add_subplot(1,5,num)
    num+=1
    img=cv2.resize(cv2.imread("./Data/images/"+test_img[0]),(224,224))
    data,preds=predict_parking(img)
    img=show_details(cv2.cvtColor(img,cv2.COLOR_BGR2RGB),data,preds,1)
    ax.imshow(img)
    print(test_img[1][-1])

```

```
['1', '1', '1', '1', '0']
['0', '0']
['0', '0']
['1']
['1', '1']
```

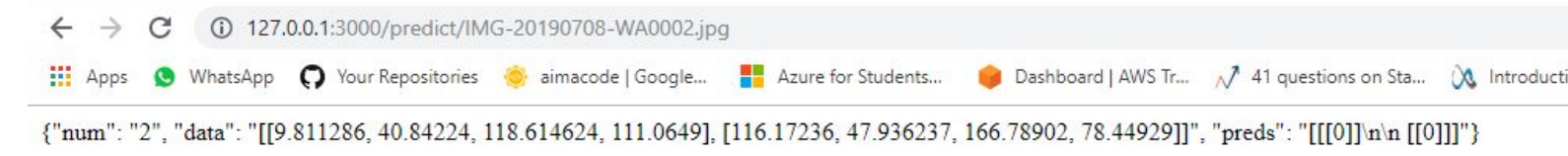


Correct predictions



Incorrect Predictions

Local API response



Api resopnse

RESULTS AND DISCUSSIONS

We have developed an Illegal Parking detection prototype which is going to be integrated with the web application for Mekvahan . It is a website that handles multiple automobile service .

All the previous illegal parking detection work is done using a video and we have developed a prototype which tries to predict using an image .

Here the term illegally parked means if the vehicle may cause any trouble due to the state of parking it is in or the location at which it is parked . Hence the model takes the background details into consideration.

Data consists of images taken from Shastri park and CP.

The model was trained on 5235 images and were tested on 269 images.

Classification report:

Category	precision	recall	f1-score	support
0 (Legally)	0.77	0.84	**0.80**	**132**
1 (Illegally)	0.83	0.75	**0.79**	**137**
accuracy			**0.80**	**269**
macro avg	0.80	0.80	**0.80**	**269**
weighted avg	0.80	0.80	**0.80**	**269**

FEATURES OF THIS APPLICATION

- This will be beneficial for not just the traffic police but also the common people to check and report an illegally parked car in order to fine the owner for the reckless parking.
- It is a cost effective way of categorizing whether the car should be fined or not.
- It requires only the image of the parked car in order to classify it as legal or illegal which makes it robust and easily manageable.
- It will create a sense of awareness and responsibility among the citizens in order to park their cars properly and also for the government to make more and more legal parking spots.

The main idea behind this project is to handle this growing problem of illegal parking using Artificial Intelligence.

We've tried applying modern Deep Learning approach in order to tell whether the parked car is legal or illegal.Our model will classify the cars from the images clicked by the users into two different categories that is, legally and illegally parked cars using deep learning.
Using AI based approach to detect illegal parking of vehicles (Cars) from an image

Our model will receive an image of parked car through the user of the app , then our model will try to predict the status of parking based on the background details (Traffic Signs , Crowded (market) area, Open field, Traffic , On road , number of side cars etc.).

Our model will do the prediction for each car that is present in the image.
With the high percentage of vehicle ownership in the India, parking has become a conflicting and confusing situation for the people.

Cars are parked everywhere in a reckless manner around us which has become the root cause of all the major traffic jams.

With the growing number of vehicles, this problem of illegal parking is increasing day by day.

Even if the cars are parked illegally there are no measurements taken in order to curb it. Not much people are fined and therefore keeps on parking illegally.

People don't know about the rules that whether to park a car at a particular location or not and in turn park anywhere they like causing traffic jams and even accidents in some cases.

REFERENCES

Various frameworks/libraries are used in this project.

- Tensorflow http://tensorflow.org/
- Keras https://www.keras.io/
- Numpy https://www.numpy.org/
- Pandas https://pandas.pydata.org/
- Open CV https://www.opencv.org/
- SSD https://github.com/tensorflow/models/tree/master/research/object_detection/
- Matplot https://matplot.org/

Apart from these resources other resources are also used for various purposes

- Stack Overflow https://stackoverflow.com/ (For Error Solving)
- arXiv https://arxiv.org › stat (For research)
- Medium https://medium.com/ (For research)