PROTOTYPE

(Smart Mall)

The prototype which we have designed is the smart parking which is to be used in a smart mall. The system works in a way that as soon as your vehicle arrives, the camera would take the photo of the number plate and provide it to the machine learning model trained for our prototype. The model accepts the image then resizes it to 340 X 110 pixels (suited for our model to get better results) and automatically allocates the area for parking and enters the car number in the database.

Images of some car license plates:

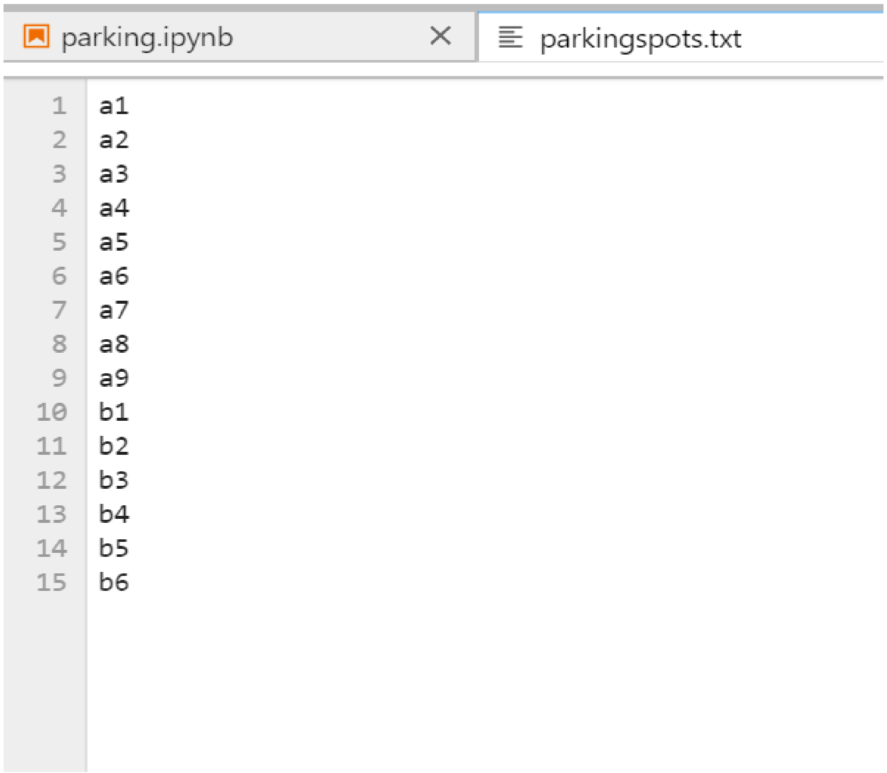
IND
HR26DK8337

IND
DL 3C AM 0857

Jeep
KL 65 K 7111

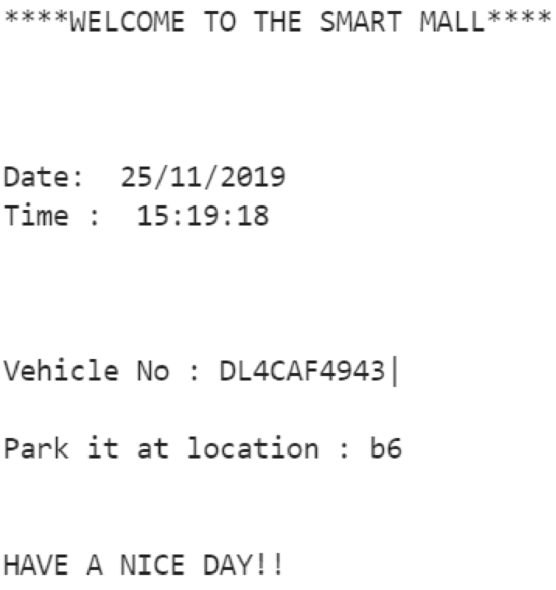
The parking locations provided:

These are the assumed parking spots available in the parking.



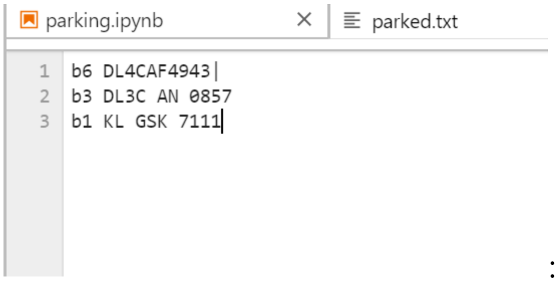
```
a1
a2
a3
a4
a5
a6
a7
a8
a9
b1
b2
b3
b4
b5
b6
```

Output provided by the model on encountering the first image:



```
****WELCOME TO THE SMART MALL****



Date:  25/11/2019
Time :  15:19:18



Vehicle No : DL4CAF4943|

Park it at location : b6


HAVE A NICE DAY!!
```

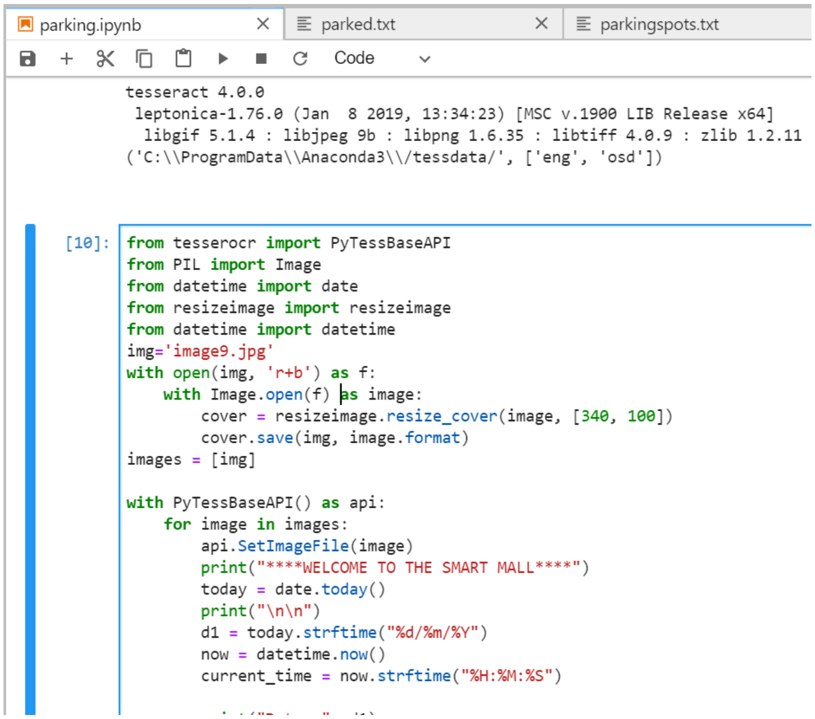
Database where the parked cars are stored:



```
b6 DL4CAF4943|
b3 DL3C AN 0857
b1 KL GSK 7111
```

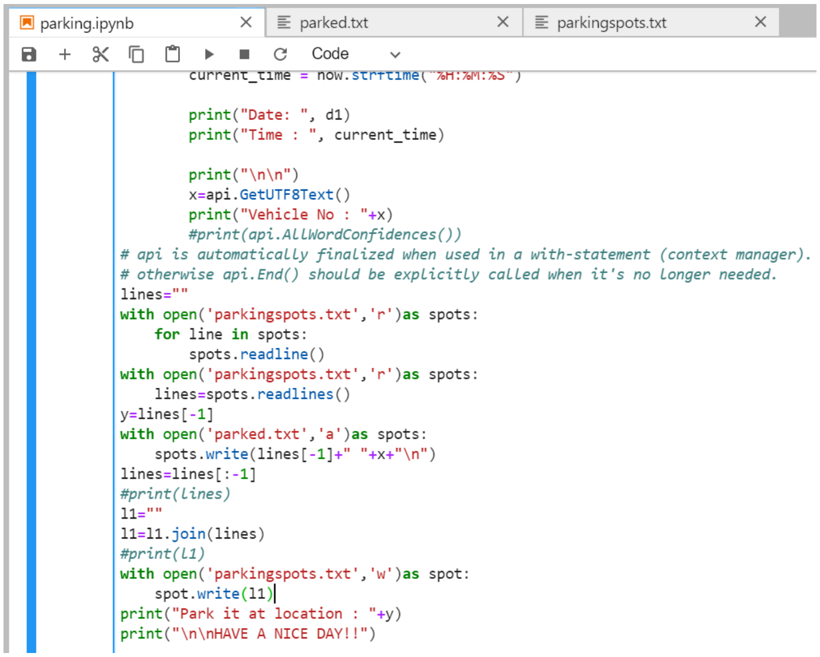
:

Images of the Machine learning model:



```
tesseract 4.0.0
 leptonica-1.76.0 (Jan  8 2019, 13:34:23) [MSC v.1900 LIB Release x64]
  libgif 5.1.4 : libjpeg 9b : libpng 1.6.35 : libtiff 4.0.9 : zlib 1.2.11
('C:\\ProgramData\\Anaconda3\\/tessdata/', ['eng', 'osd'])
```

```
[10]: from tesserocr import PyTessBaseAPI
      from PIL import Image
      from datetime import date
      from resizeimage import resizeimage
      from datetime import datetime
      img='image9.jpg'
      with open(img, 'r+b') as f:
          with Image.open(f) as image:
              cover = resizeimage.resize_cover(image, [340, 100])
              cover.save(img, image.format)
      images = [img]

      with PyTessBaseAPI() as api:
          for image in images:
              api.SetImageFile(image)
              print("****WELCOME TO THE SMART MALL****")
              today = date.today()
              print("\n\n")
              d1 = today.strftime("%d/%m/%Y")
              now = datetime.now()
              current_time = now.strftime("%H:%M:%S")
```



```
        current_time = now.strftime("%H:%M:%S")

        print("Date: ", d1)
        print("Time : ", current_time)

        print("\n\n")
        x=api.GetUTF8Text()
        print("Vehicle No : "+x)
        #print(api.AllWordConfidences())
# api is automatically finalized when used in a with-statement (context manager).
# otherwise api.End() should be explicitly called when it's no longer needed.
lines=""
with open('parkingspots.txt','r')as spots:
    for line in spots:
        spots.readline()
with open('parkingspots.txt','r')as spots:
    lines=spots.readlines()
y=lines[-1]
with open('parked.txt','a')as spots:
    spots.write(lines[-1]+" "+x+"\n")
lines=lines[:-1]
#print(lines)
l1=""
l1=l1.join(lines)
#print(l1)
with open('parkingspots.txt','w')as spot:
    spot.write(l1)
print("Park it at location : "+y)
print("\n\nHAVE A NICE DAY!!")
```

Summary:

As soon as the license plate is encountered by the model it reads the text from the image and out of the parking spots available, hands over a parking spot to it and enters the given spot and car number in another database. Thus, a smart way of parking where no need to look for parking spots and also it is easy to search for the car if forgotten the parking spot.