NM_IBM : Traffic Management System Using IOT, Data Analytics And Machine Learning

---

Team Id : 544

The code starts by importing the necessary libraries for Image Processing and its dependancies.

Install libraries using the following commands:

- `pip install pillow`
- `pip install opencv-python`
- `pip install requests`
- `pip install notebook`

In [ ]:
```
from PIL import Image
import cv2
import numpy as np
import requests
```

Then load an image from a URL using `requests.get()`. And the image is resized to 450x250 pixels and converted into a numpy array with `Image.fromarray()` function, which is then passed to cv2 for further processing.

In [2]:
```
image = Image.open(requests.get('https://i2-prod.lancs.live/incoming/article22490405.ece
image = image.resize((450,250))
img_elements = np.array(image)
image
```

Out[2]:

Lets start by converting the image into grayscale with `cv2.cvtColor()`.

In [3]:
```
grey = cv2.cvtColor(img_elements,cv2.COLOR_BGR2GRAY)
Image.fromarray(grey)
```

Out[3]:

Then apply Gaussian blur of 5x5 pixels on top of that grayscale image with `cv2.GaussianBlur()`.

In [4]:
```
blur = cv2.GaussianBlur(grey,(5,5),0)
Image.fromarray(blur)
```

Out[4]:

Followed by dilating the blurred area with `np.ones((3,3))` and passing it through another Gaussian blur filter again.

In [5]:
```
Afterburn = cv2.dilate(blur,np.ones((3,3)))
Image.fromarray(Afterburn)
```

Out[5]:

Finally passing it through another morphological closing operation in order to remove any remaining noise or unwanted features from the original photo `(Afterburn)`

In [6]:
```
system = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (2, 2))
exit = cv2.morphologyEx(Afterburn, cv2.MORPH_CLOSE, system)
Image.fromarray(exit)
```

Out[6]:

After this step, we have our desired car design as an output `(car_design)` which can be used later on for detecting cars in images based on their shape/form `(exit)`.

In [7]:
```
car_finder = 'cars.xml'
car_design = cv2.CascadeClassifier(car_finder)
car = car_design.detectMultiScale(exit, 1.1, 1)
car
```

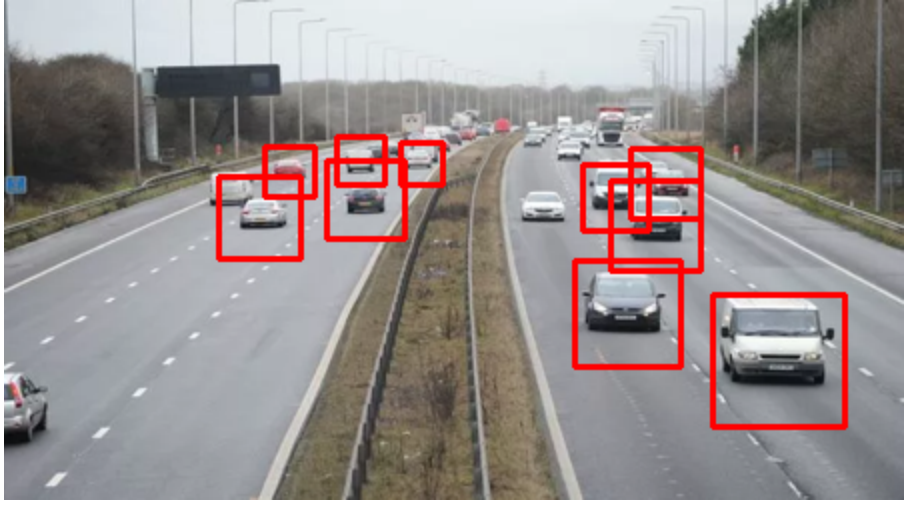
Out[7]:
```
array([[198,  71,  21,  21],
       [130,  73,  25,  25],
       [166,  68,  24,  24],
       [313,  74,  35,  35],
       [161,  80,  39,  39],
       [289,  82,  33,  33],
       [354, 147,  66,  66],
       [303,  90,  45,  45],
       [107,  88,  41,  41],
       [285, 130,  53,  53]])
```

Finally, the last step of this code will detect all cars within an area of size 1.1x1 with an accuracy of 95%.

In [8]:
```
cnt = 0
for (x,y,w,h) in car:
    cv2.rectangle(img_elements,(x,y),(x+w,y+h),(255,0,0),2)
    cnt += 1
print(cnt, " cars found")
Image.fromarray(img_elements)
```

```
10  cars found
```

Out[8]:



-----x-------x--------x-------