1. Training and Testing :

Input : training\_chars.png

Output : classifications.txt, flattened\_image.txt

Testing was done using several datasets and as mentioned in the dataset description. For eg. Refer folder *car\_image , kaggle dataset link in dataset description.*

Sources:

[tushar1210/OCR-KNN: Optical Character Recognition using KNN for my IP project](https://github.com/tushar1210/OCR-KNN)

[OpenCV Intro to Character Recognition and Machine Learning with KNN](https://www.youtube.com/watch?v=6FzlF9qf960)

1. Bounding Box : **Bounding Box.ipynb**

Input: Test Video.mp4

Output1 : Video splitted into frames and saved in folder *photos.*

Output2 : Region of interest recognized from each frame and saved in folder *BoundedBox*.

Sources:

[Python | Program to extract frames using OpenCV](https://www.geeksforgeeks.org/python-program-extract-frames-using-opencv/)

1. Optical Character recognition using KNN : **Video to License Plate.ipynb**

Input: Test Video.mp4 (The cropped image dataset can also be used)

Output1 : Cropped Image of license plate

Output2 : Recognized characters in the output console.

Sources:

[How to Recognize License Plate with Python OpenCV Machine learning?](https://www.youtube.com/watch?v=AifDb8oG8Kc)

[License plate detection & recognition using opencv & pytesseract | The Legendary Outlier](https://www.youtube.com/watch?v=GXyLSx8l9gM)