# -\*- coding: utf-8 -\*-

"""Copy of Untitled16.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/1fKNcIXY00MXzvs3UpsFWDTrzvyVCtcHr

"""

!pip install opencv-python

!pip install pytesseract

!git clone https://github.com/kennethleungty/Car-Plate-Detection-OpenCV-TesseractOCR

# Commented out IPython magic to ensure Python compatibility.

# Import dependencies

import numpy as np

import matplotlib.pyplot as plt

# %matplotlib inline

import cv2 # This is the OpenCV Python library

import pytesseract # This is the TesseractOCR Python library

def enlarge\_plt\_display(image, scale\_factor):

width = int(image.shape[1] \* scale\_factor / 100)

height = int(image.shape[0] \* scale\_factor / 100)

dim = (width, height)

plt.figure(figsize = dim)

plt.axis('off')

plt.imshow(image)

# Read car image and convert color to RGB

carplate\_img = cv2.imread('/content/car.jpg')

carplate\_img\_rgb = cv2.cvtColor(carplate\_img, cv2.COLOR\_BGR2RGB)

enlarge\_plt\_display(carplate\_img\_rgb, 1.2)



carplate\_haar\_cascade = cv2.CascadeClassifier('/content/Car-Plate-Detection-OpenCV-TesseractOCR/haar\_cascades/haarcascade\_russian\_plate\_number.xml')

# Setup function to detect car plate

def carplate\_detect(image):

carplate\_overlay = image.copy()

carplate\_rects = carplate\_haar\_cascade.detectMultiScale(carplate\_overlay,scaleFactor=1.1, minNeighbors=5)

for x,y,w,h in carplate\_rects:

cv2.rectangle(carplate\_overlay, (x,y), (x+w,y+h), (255,0,0), 5)

return carplate\_overlay

detected\_carplate\_img = carplate\_detect(carplate\_img\_rgb)

enlarge\_plt\_display(detected\_carplate\_img, 1.2)



def carplate\_extract(image):

carplate\_rects = carplate\_haar\_cascade.detectMultiScale(image,scaleFactor=1.1, minNeighbors=5)

for x,y,w,h in carplate\_rects:

carplate\_img = image[y+15:y+h-10 ,x+15:x+w-20]

return carplate\_img

# Enlarge image for further image processing later on

def enlarge\_img(image, scale\_percent):

width = int(image.shape[1] \* scale\_percent / 100)

height = int(image.shape[0] \* scale\_percent / 100)

dim = (width, height)

resized\_image = cv2.resize(image, dim, interpolation = cv2.INTER\_AREA)

return resized\_image

# Display extracted car license plate image

carplate\_extract\_img = carplate\_extract(carplate\_img\_rgb)

carplate\_extract\_img = enlarge\_img(carplate\_extract\_img, 150)

plt.imshow(carplate\_extract\_img);



# Convert image to grayscale

carplate\_extract\_img\_gray = cv2.cvtColor(carplate\_extract\_img, cv2.COLOR\_RGB2GRAY)

plt.axis('off')

plt.imshow(carplate\_extract\_img\_gray, cmap = 'gray');

