```
!nvcc --version
```

```
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2022 NVIDIA Corporation
Built on Wed_Sep_21_10:33:58_PDT_2022
Cuda compilation tools, release 11.8, V11.8.89
Build cuda_11.8.r11.8/compiler.31833905_0
```

```
!pip install easyocr
!pip install imutils
!pip insatt opencv-python-hedless==4.1.2.30
!pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118
```

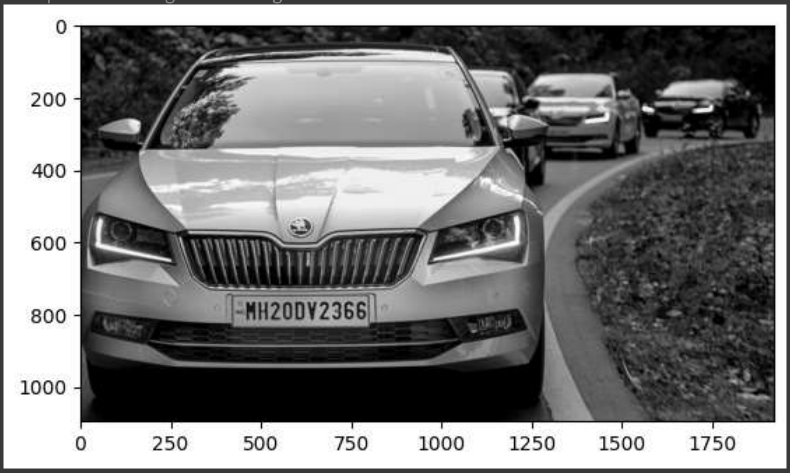
```
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from easyocr) (1.11.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from easyocr) (1.23.5)
Requirement already satisfied: Pillow in /usr/local/lib/python3.10/dist-packages (from easyocr) (9.4.0)
Requirement already satisfied: scikit-image in /usr/local/lib/python3.10/dist-packages (from easyocr) (0.19.3)
Collecting python-bidi (from easyocr)
  Downloading python_bidi-0.4.2-py2.py3-none-any.whl (30 kB)
Requirement already satisfied: PyYAML in /usr/local/lib/python3.10/dist-packages (from easyocr) (6.0.1)
Requirement already satisfied: Shapely in /usr/local/lib/python3.10/dist-packages (from easyocr) (2.0.1)
Collecting pyclipper (from easyocr)
  Downloading pyclipper-1.3.0.post5-cp310-cp310-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (908 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 908.3/908.3 kB 40.8 MB/s eta 0:00:00
Collecting ninja (from easyocr)
  Downloading ninja-1.11.1-py2.py3-none-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (145 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 146.0/146.0 kB 17.5 MB/s eta 0:00:00
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from torchvision>=0.5->easyocr) (2.31.
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch->easyocr) (3.12.2)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10/dist-packages (from torch->easyocr) (4.5.0)
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch->easyocr) (1.12)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch->easyocr) (3.1)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch->easyocr) (3.1.2)
Requirement already satisfied: triton==2.0.0 in /usr/local/lib/python3.10/dist-packages (from torch->easyocr) (2.0.0)
Requirement already satisfied: cmake in /usr/local/lib/python3.10/dist-packages (from triton==2.0.0->torch->easyocr) (3.27
Requirement already satisfied: lit in /usr/local/lib/python3.10/dist-packages (from triton==2.0.0->torch->easyocr) (16.0.6
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from python-bidi->easyocr) (1.16.0)
Requirement already satisfied: imageio>=2.4.1 in /usr/local/lib/python3.10/dist-packages (from scikit-image->easyocr) (2.31
Requirement already satisfied: tifffile>=2019.7.26 in /usr/local/lib/python3.10/dist-packages (from scikit-image->easyocr)
Requirement already satisfied: PyWavelets>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-image->easyocr) (
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from scikit-image->easyocr) (23
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch->easyocr) (2
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->torchvis
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->torchvision>=0.5->ea
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->torchvision>=
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->torchvision>=
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-packages (from sympy->torch->easyocr) (1.3.0
Installing collected packages: pyclipper, ninja, python-bidi, easyocr
Successfully installed easyocr-1.7.1 ninja-1.11.1 pyclipper-1.3.0.post5 python-bidi-0.4.2
Requirement already satisfied: imutils in /usr/local/lib/python3.10/dist-packages (0.5.4)
ERROR: unknown command "insatt" - maybe you meant "install"
Looking in indexes: https://download.pytorch.org/whl/cu118
Requirement already satisfied: torch in /usr/local/lib/python3.10/dist-packages (2.0.1+cu118)
Requirement already satisfied: torchvision in /usr/local/lib/python3.10/dist-packages (0.15.2+cu118)
Requirement already satisfied: torchaudio in /usr/local/lib/python3.10/dist-packages (2.0.2+cu118)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch) (3.12.2)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10/dist-packages (from torch) (4.5.0)
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch) (1.12)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch) (3.1)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch) (3.1.2)
Requirement already satisfied: triton==2.0.0 in /usr/local/lib/python3.10/dist-packages (from torch) (2.0.0)
Requirement already satisfied: cmake in /usr/local/lib/python3.10/dist-packages (from triton==2.0.0->torch) (3.27.4.1)
Requirement already satisfied: lit in /usr/local/lib/python3.10/dist-packages (from triton==2.0.0->torch) (16.0.6)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from torchvision) (1.23.5)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from torchvision) (2.31.0)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in /usr/local/lib/python3.10/dist-packages (from torchvision) (9.4.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch) (2.1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->torchvis
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->torchvision) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->torchvision)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->torchvision)
```

```
import cv2
from matplotlib import pyplot as plt
import numpy as np
import imutils
import easyocr
```

Convert Colored Image into Grayscale Format

```
# Taking input of cars image
img = cv2.imread("/content/india-skoda-license-plate.jpg")
gray = cv2.cvtColor(img , cv2.COLOR_BGR2GRAY)
plt.imshow(cv2.cvtColor(gray, cv2.COLOR_BGR2RGB))
```

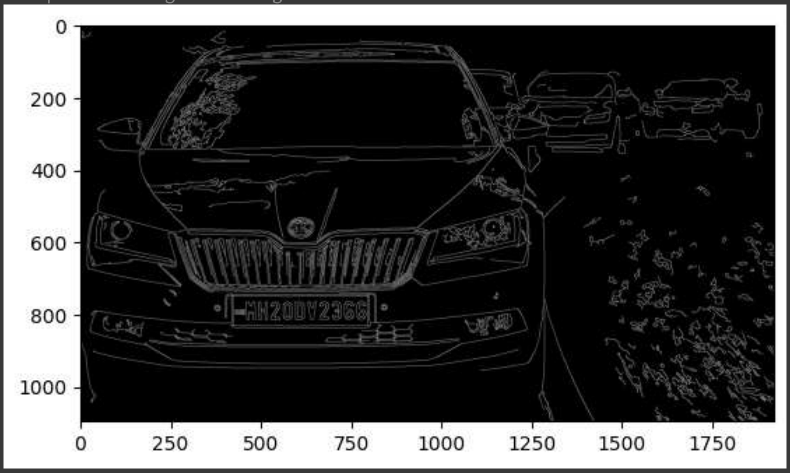
```
<matplotlib.image.AxesImage at 0x7dc72f636440>
```



Apply Filter and Edge Detection

```
bfilter = cv2.bilateralFilter(gray, 11,17,17)
edged = cv2.Canny(bfilter, 30 ,200)
plt.imshow(cv2.cvtColor(edged, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x7dc72f532650>
```



Find Contours and Apply Mask

```
keypoints = cv2.findContours(edged.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
contours = imutils.grab_contours(keypoints)
contours = sorted(contours, key = cv2.contourArea, reverse=True)[:10]
```

```
location = None
for contour in contours:
    approx = cv2.approxPolyDP(contour,50,True)
    if len(approx)==4:
        location = approx
        break
```

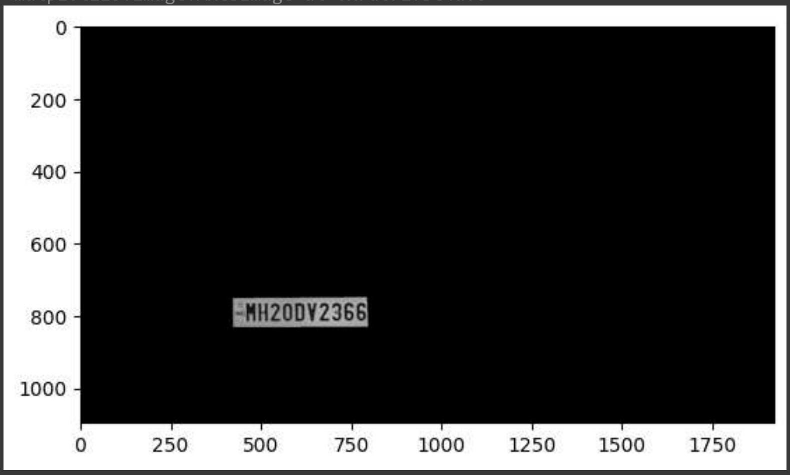
```
location
```

```
array([[[421, 753]],

       [[421, 832]],

       [[794, 831]],

       [[791, 750]]], dtype=int32)
```

```
mask  = np.zeros(gray.shape, np.uint8)
new_image = cv2.drawContours(mask, [location], 0, 255, -1)
new_image = cv2.bitwise_and(img, img, mask=mask)

```

```
plt.imshow(cv2.cvtColor(new_image, cv2.COLOR_BGR2RGB))
```

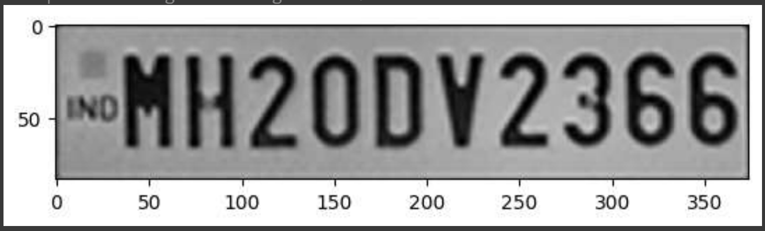
```
<matplotlib.image.AxesImage at 0x7dc72f5e4a00>
```



```
(x,y) = np.where(mask==255)
(x1,y1) = (np.min(x), np.min(y))
(x2, y2) = (np.max(x), np.max(y))
cropped_image = gray[x1:x2+1, y1:y2+1]
```

```
plt.imshow(cv2.cvtColor(cropped_image, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x7dc72f47f7f0>
```



Easy OCR to read Text

```
reader = easyocr.Reader(['en'])
result = reader.readtext(cropped_image)
result
```

```
WARNING:easyocr.easyocr:Neither CUDA nor MPS are available - defaulting to CPU. Note: This module is much faster with a GPU.
[([[0, 1], [374, 1], [374, 79], [0, 79]], '~HHZODV2366 ', 0.44101918310346316)]
```

```
text = result[0][-2]
font = cv2.FONT_HERSHEY_SIMPLEX
res = cv2.putText(img, text, (approx[0][0][0], approx[1][0][1] + 60), font, 1, (0, 255, 0), 2, cv2.LINE_AA, False)
res = cv2.rectangle(img, tuple(approx[0][0]), tuple(approx[2][0]), (0, 255, 0), 3)
plt.imshow(cv2.cvtColor(res, cv2.COLOR_BGR2RGB))

```

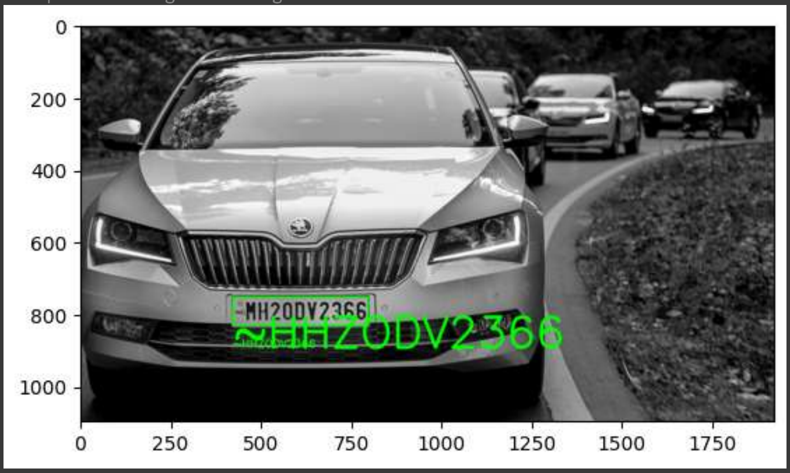
```
<matplotlib.image.AxesImage at 0x7dc72f3267a0>
```

```
NumberPlate = result[0][1]
print("Vehicle Number : ",NumberPlate)
```

```
Vehicle Number :  ~HHZODV2366
```

```
text = result[0][-2]
font = cv2.FONT_HERSHEY_SIMPLEX
font_scale = 4.0  # You can adjust this value to increase or decrease the font size
font_thickness = 7  # You can adjust this value to change the font thickness

# Calculate the position for the text
text_position = (approx[0][0][0], approx[1][0][1] + 60)

# Draw the text with the specified font size and thickness
res = cv2.putText(img, text, text_position, font, font_scale, (0, 255, 0), font_thickness, cv2.LINE_AA, False)

# Draw the rectangle
res = cv2.rectangle(img, tuple(approx[0][0]), tuple(approx[2][0]), (0, 255, 0), 3)

# Display the image
plt.imshow(cv2.cvtColor(res, cv2.COLOR_BGR2RGB))

```

```
<matplotlib.image.AxesImage at 0x7dc72f34b370>
```



Storing the number of vehicle in a CSV file

```
import datetime
import pytz

# Specify the timezone for India
india_timezone = pytz.timezone('Asia/Kolkata')

# Get the current date and time in India
current_date_time_in_india = datetime.datetime.now(india_timezone)

# Format the date and time
formatted_date_time = current_date_time_in_india.strftime("%Y-%m-%d %H:%M:%S")

# Print the current date and time in India
print("Current date and time in India:", formatted_date_time)

```

```
Current date and time in India: 2023-09-16 00:15:07
```

```
import datetime
import csv
import pytz

# Specify the timezone for India
india_timezone = pytz.timezone('Asia/Kolkata')
```

```
# Your calculator code here
result = NumberPlate  # Replace this with your calculation

# Get the current date and time in India's timezone
current_datetime = datetime.datetime.now(india_timezone)

# Get the current date in YYYY-MM-DD format
current_date = current_datetime.strftime('%Y-%m-%d')

# Format the time to include only hour and minute
current_time = current_datetime.strftime('%H:%M')

# Create a CSV file name with the current date and time
file_name = f"NumberPlate_on_{current_date}.csv"

# Create a list with the data you want to write to the CSV file
data_to_write = [("Result", "Date", "Time"), (result, current_date, current_time)]

# Open the CSV file in write mode
with open(file_name, "w", newline='') as csv_file:
    csv_writer = csv.writer(csv_file)

    # Write the data to the CSV file
    csv_writer.writerows(data_to_write)

print(f"Result has been saved to {file_name}")

# Now, you can read the CSV file
try:
    # Open the CSV file in read mode
    with open(file_name, mode='r') as csv_file:
        # Create a CSV reader object
        csv_reader = csv.reader(csv_file)

        # Iterate through each row in the CSV file
        for row in csv_reader:
            print(row)
except FileNotFoundError:
    print(f"The file '{file_name}' was not found.")

```

```
Result has been saved to NumberPlate_on_2023-09-16.csv
['Result', 'Date', 'Time']
['~HHZODV2366 ', '2023-09-16', '00:39']
```

Colab paid products - Cancel contracts here

✓ 0s completed at 12:39 AM