



PENDETEKSI PLAT NOMOR

Husnul Latifah

17/409375/PA/17682

Ratih Suci Lestari

20/456556/PA/19743

Datu Maulana Ahmad

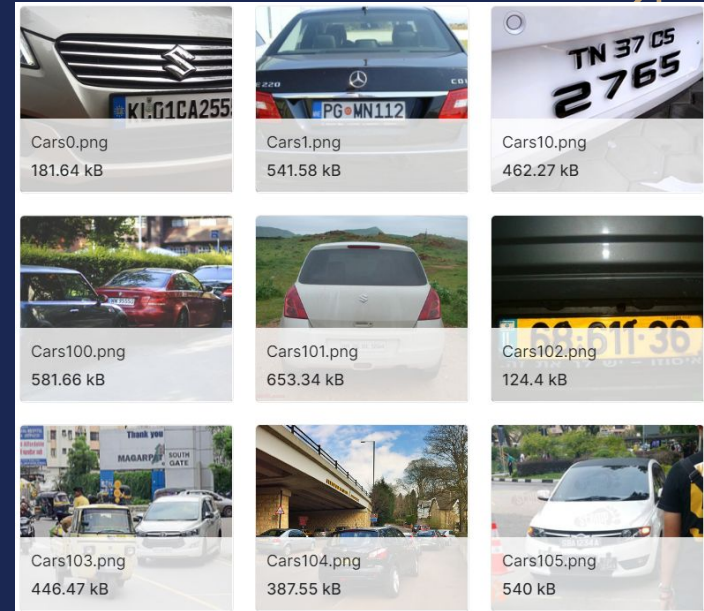
20/462077/PA/20049

Khairunas Rhamadhani Wiasanto

20/462085/PA/20057

Dataset

Dataset yang digunakan pada tugas ini Car License Plate Detection yang berasal dari kaggle.



Library

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import cv2
5 import os
6 import pathlib
```



Pre-processing

Tahap ini dilakukan untuk mengubah citra RGB menjadi BGR, agar citra kembali ke warna aslinya, dan setelah itu citra diubah menjadi grayscale.

```
1 image = cv2.imread(path)
2 rgb = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
3 plot_image = image_read(image, rgb, title1 = "RGB", title2 = "Original")
```

```
gray = cv2.cvtColor(rgb, cv2.COLOR_BGR2GRAY)
image_read(rgb, gray, title1 = "RGB", title2 = "Gray")
```



Peningkatan Kualitas Citra

Pada tahap ini digunakan smoothing metode Gaussian Smoothing yang memiliki fungsi untuk menyamarkan noise pada sehingga dapat memudahkan proses dalam pendeteksian

```
1 #kernel gaussian disesuaikan nilainya  
2 #makin besar nilainya makin blur  
3 smooth = cv2.GaussianBlur(gray,(1,1),cv2.BORDER_DEFAULT)  
4 image_read(gray, smooth, title1 = "Grayscale", title2 = "Gaussian Smoothing")
```



Segmentasi

Pada tahap ini digunakan menggunakan metode otsu thresholding yang memiliki fungsi untuk memisahkan antara bagian latar dan objek dari citra yang berbentuk grayscale.

```
1 otsu_threshold, sigma_2_T, th_image = otsu_image_thresholding(smooth)
2 print("Otsu Image otsu_threshold: ", otsu_threshold)
3 image_read(smooth, th_image, title1 = "Gaussian Smoothing", title2 = "Otsu Thresholding")
```

Otsu Image otsu_threshold: 93

Gaussian Smoothing



Otsu Thresholding

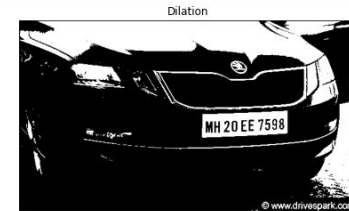
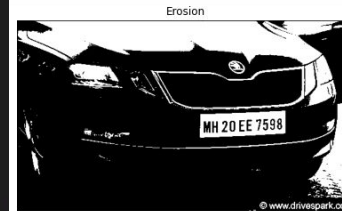


Operasi Morfologi

Setelah dilakukan otsu thresholding kemudian dilakukan pendekatan dengan erosion, dilatation, closing dan opening.



```
1  #nilai kernel disesuaikan
2  kernel = np.ones((1, 1), np.uint8)
3
4  erosion = cv2.erode(th_image, kernel, iterations=1)
5  dilation = cv2.dilate(th_image, kernel, iterations=1)
6
7  closing = cv2.morphologyEx(th_image, cv2.MORPH_CLOSE, kernel, iterations=1)
8  opening = cv2.morphologyEx(th_image, cv2.MORPH_OPEN, kernel, iterations=1)
9
10 image_read(erosion, dilation, title1 = "Erosion", title2 = "Dilation")
11 image_read(opening, closing, title1 = "Opening", title2 = "Closing")
```

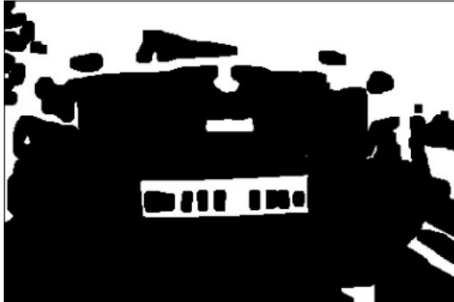


Contoh Kasus

Gaussian Smoothing



Erosion



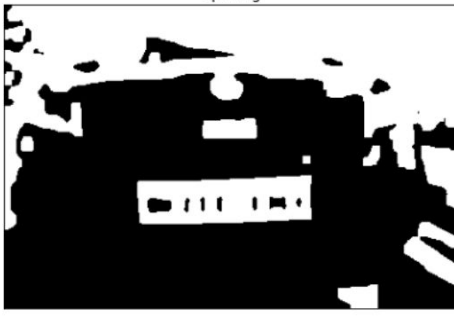
Dilation



Otsu Thresholding



Opening



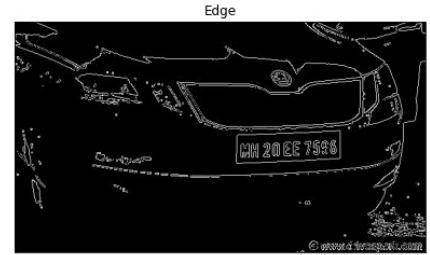
Closing



Ekstraksi Fitur

Pada tahapan ini digunakan metode edge based yang berfungsi untuk mendeteksi tepi-tepi dari citra dengan metode canny sehingga huruf dan angka dapat terdeteksi.

```
1 edge = cv2.Canny(erosion, 70, 400)
2 image_read(erosion, edge, title1 = "Erosion", title2 = "Edge")
```



Deteksi

Pada tahap deteksi digunakan metode representasi citra digital kontur yang berfungsi sebagai penghubung antara titik-titik yang sama dan dapat memberikan informasi relief.

Selanjutnya kontour-kontour yang ada direduksi untuk mencari kontour yang dominan.

```
contours, new = cv2.findContours(edge.copy(), cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)
image_copy = rgb.copy()
_ = cv2.drawContours(image_copy, contours, -1, (255, 0, 0), 2)
image_read(edge, image_copy, title1 = "Edge", title2 = "Contours")
```



```
1 contours = sorted(contours, key = cv2.contourArea, reverse = True)[:4]
2
3 image_reduced = edge.copy()
4 _ = cv2.drawContours(image_reduced, contours, -1, (255, 0, 0), 2)
5
6 image_read(image_copy, image_reduced, title1 = "Original", title2 = "Reduced")
```



Deteksi

Pada tahap ini menggunakan metode edge counting dengan perulangan, untuk mencari titik-titik yang disinyalir sebagai plat kendaraan.

```
1 image_read(plate, rgb, title1 = "Plate", title2 = "Original")
```



```
1 plate = None
2
3 for i in contours:
4     a = cv2.arcLength(i, True)
5     edge_count = cv2.approxPolyDP(i, 0.005 * a, True)
6     print('edge_count: ', edge_count)
7     if len(edge_count) >= 4 and len(edge_count) < 6:
8         x, y, w, h = cv2.boundingRect(i)
9         plate = image[y:y+h, x:x+w]
10        break
11
12 cv2.imwrite("plate.png", plate)
```

```
[> edge_count: [[[475 161]]
[[326 166]]
[[323 206]]
[[332 208]]
[[475 200]]]
True
```

Hasil Deteksi

Plate



Original



Plate



Original



Plate



Original



Plate



Original



Terima Kasih

