**LAPORAN**

**PEMROGRAMAN CITRA DAN VISI KOMPUTER**

**“ DETEKSI PLATE MOBIL DENGAN OPEN CV DAN TESSERACT “**



Disusun oleh:

|  |  |  |
| --- | --- | --- |
| Aryo Bagus Kusumadewa Tutuko | TI – 3A | 2041720232 |
| Maulana Syarief Hidayahtullah | TI – 3A | 2041720190 |
| Nafarul Hamka | TI – 3A | 2041720101 |

**D4 TEKNIK INFORMATIKA**

**TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2023**

**Laporan**

1. **Latar Belakang**
   1. **Masalah**

Deteksi plat nomor mobil merupakan tantangan yang kompleks dalam pengembangan sistem pengenalan plat nomor. Berbagai faktor seperti variasi bentuk dan ukuran plat nomor, kondisi pencahayaan yang berbeda, gangguan visual, kecepatan dan pergerakan kendaraan, serta kepadatan lalu lintas mempengaruhi akurasi deteksi. Namun, melalui penelitian dan pengembangan teknologi pengolahan citra dan kecerdasan buatan, upaya terus dilakukan untuk meningkatkan kualitas deteksi plat nomor mobil dalam berbagai kondisi.

* 1. **Solusi**

Deteksi plate mobil dengan open cv dan tesseract adalah salah satu solusi untuk mendeteksi plate nomor pada mobil. Metode ini menggunakan open cv dan tesseract untuk memproses gambar dan pengenalan karakter pada citra pada plat nomor yang sudah dipotong. Dengan menggabungkan kedua library ini kita dapat mengimplementasikan system deteksi dan pengenalan plat nomor mobil secara otomatis.

* 1. **Metode**

OpenCV (Open Source Computer Vision) adalah library populer yang digunakan untuk pemrosesan citra dan visi komputer. OpenCV menyediakan berbagai fungsi dan algoritma yang berguna untuk deteksi, ekstraksi fitur, segmentasi, dan manipulasi gambar. Dalam konteks deteksi plat nomor mobil, OpenCV dapat digunakan. Tesseract adalah mesin pengenalan karakter optik (OCR) yang dikembangkan oleh Google. Tesseract memiliki kemampuan untuk membaca dan mengenali karakter pada gambar. Jadi, OpenCV digunakan untuk tahapan awal seperti pra-pemrosesan gambar, deteksi objek, dan pemrosesan citra, sedangkan Tesseract digunakan untuk tahapan pengenalan karakter pada citra plat nomor yang sudah dipotong. Dengan menggabungkan kedua library.

1. **Praktikum**
   1. **Library**

Library yang digunakan adalah :

import cv2

import numpy as np

import matplotlib.pyplot as plt

import pytesseract

import os

1. Library CV2 digunakan untuk memanipulasi gambar, melakukan operasi seperti membaca gambar, melakukan filter, pemrosesan citra.
2. Library Numpy digunakan untuk melakukan operasi numerik pada array, dan dalam tugas ini ia berfungsi untuk mengolah gambar dalam bentuk array.
3. Library Matplotlib digunakan untuk visualisasi data dalam hal ini untuk menampilkan gambar hasil pemrosesan
4. Library pytesseract adalah mesin OCR open source yang digunakan untuk mengenali dan ekstraksi gambar.
5. Library os adalah modul python yang menyediakan fugsi untuk berinteraksi dengan system operasi, dalam masalah ini ia berfungsi untuk berinteraksi dengan file system missal akses directori atau interaksi dengan file lain
   1. **Preprocessing**

Kode yang digunakan adalah :

def preprocessing(img, h1, h2, w1, w2):

    # get roi & resize

    y1, y2, x1, x2 = int(h1), int(h2), int(w1), int(w2)

    roi = img[y1:y2, x1:x2]

    scale = 300/roi.shape[0] # roi.shape[0] merupakan h image roi

    roi = cv2.resize(roi, (0,0), fx=scale, fy=scale)

    #convert to gray -> binary

    gray = cv2.cvtColor(roi, cv2.COLOR\_BGR2GRAY)

    \_\_, thresh = cv2.threshold(gray, 0, 255, cv2.THRESH\_BINARY + cv2.THRESH\_OTSU)

    return thresh, roi, [x1, y1, x2, y2], scale

Kode diatas merupakan sebuah fungsi yang digunakan untuk melakukan pra-pemrosesan gambar sebelum dilakukan pengenalan teks menggunakan OCR (Optical Character Recognition). Fungsi pertama ini mengambil gambar input dan batas-batas region of interest (ROI) yang diinginkan, fungsi ini memotong bagian ROI dari gambar input. Selanjutnya, ROI tersebut diresize agar sesuai dengan kebutuhan, dengan menggunakan faktor skala tertentu untuk mengubahnya menjadi lebar 300 piksel dan tinggi yang sesuai dengan perbandingan aslinya. Selanjutnya, ROI yang sudah diresize dikonversi ke citra grayscale. Setelah itu, dilakukan proses thresholding pada citra grayscale menggunakan metode thresholding tertentu (dalam hal ini, metode yang digunakan adalah metode Otsu). Citra ini disimpan dalam variabel "thresh". Selain itu, fungsi ini juga mengembalikan informasi tambahan berupa koordinat batas ROI (x1, y1, x2, y2) dan faktor skala yang digunakan. Informasi ini dapat berguna dalam proses selanjutnya.

# calculate contour & filter contour

def get\_contours(thresh):

    contours, \_\_\_ = cv2.findContours(thresh, cv2.RETR\_CCOMP, cv2.CHAIN\_APPROX\_SIMPLE)

    h, w, c = img.shape

    size = h\*w

    contours = [cc for i, cc in enumerate(contours) if contour\_char\_OK(cc, size)]

    return contours

def contour\_char\_OK(cc, size=1000000):

    x, y, w, h = cv2.boundingRect(cc)

    area = cv2.contourArea(cc)

    if w < 3 or h < 5 or area < 80:

        return False

    validDimentson = w/h > 0.11 and w/h < 0.7 # filter rasio ukuran lebar / tinggi

    validAreaRatio = area/(w\*h)  > 0.1 # filter rasio luasan (luas area putih / hitam)

    return validDimentson and validAreaRatio

def sort\_contours(contours, method="left-to-right"):

    reverse = False

    i = 0

    if method == "right-to-left" or method == "bottom-to-top":

        reverse = True

    if method == "top-to-bottom" or method == "bottom-to-top":

        i = 1

    boundingBoxes = [cv2.boundingRect(cnt) for cnt in contours]

    cnts, boundingBoxes = zip(\*sorted(zip(contours, boundingBoxes), key=lambda b:b[1][i], reverse=reverse))

    return cnts, boundingBoxes

Fungsi **get\_contours(thresh)**:

Fungsi ini mengambil citra biner (thresh) sebagai input.

Menggunakan fungsi cv2.findContours(), Jika kontur tidak memenuhi syarat pada salah satu pengecekan tersebut, fungsi akan mengembalikan nilai False. Jika memenuhi semua syarat, fungsi akan mengembalikan nilai True.

Fungsi **sort\_contours(contours, method="left-to-right"):**

Fungsi ini mengambil list kontur (contours) dan metode pengurutan (method) sebagai input (dengan nilai default "left-to-right"). Fungsi ini melakukan pengurutan kontur-kontur berdasarkan metode pengurutan yang diinginkan. Metode pengurutan yang tersedia adalah "left-to-right", "right-to-left", "top-to-bottom", dan "bottom-to-top".Hasil pengurutan kontur-kontur dan bounding box-nya dikembalikan sebagai output fungsi.

* 1. **Main Program**

kode yang digunakan adalah :

img = cv2.imread("/content/data\_test/WhatsApp Image 2023-06-14 at 19.44.29.jpeg")

h, w, c = img.shape

thresh, roi, roi\_rect, ratio = preprocessing(img, h1=0.5\*h, h2=0.85\*h, w1=0.3\*w, w2=0.7\*w)

pytesseract.pytesseract.tesseract\_cmd

# get contour

contours = get\_contours(thresh)

pts = []

if len(contours) < 3:

    print("[STOP] tidak dapat mendeteksi plat nomor, contour terlalu sedikit..")

else:

    # get 4 point coordinate plate | pts = tl, tr, br, bl

    pts = get\_plate\_4\_coord(contours)

    # geometric transform & crop plate number box

    plate\_img, plate\_rect = transform(roi, pts, padding=3)

    plate\_img = np.uint8(plate\_img)

    # recognize plate character

    configuration = "--oem 1 --psm 7"

    plate\_text = pytesseract.image\_to\_string(plate\_img, config=configuration)

    print(plate\_text.strip())

    # draw label on image

    left, top, right, bottom = get\_box\_original(plate\_rect, roi\_rect, ratio)

    img = drawPred(img, plate\_text.strip(), left - 5, top - 10, right + 5, bottom + 10)

    # display result

    plt.imshow(plate\_img[:, :, ::-1])

    plt.axis('off')

    plt.show()

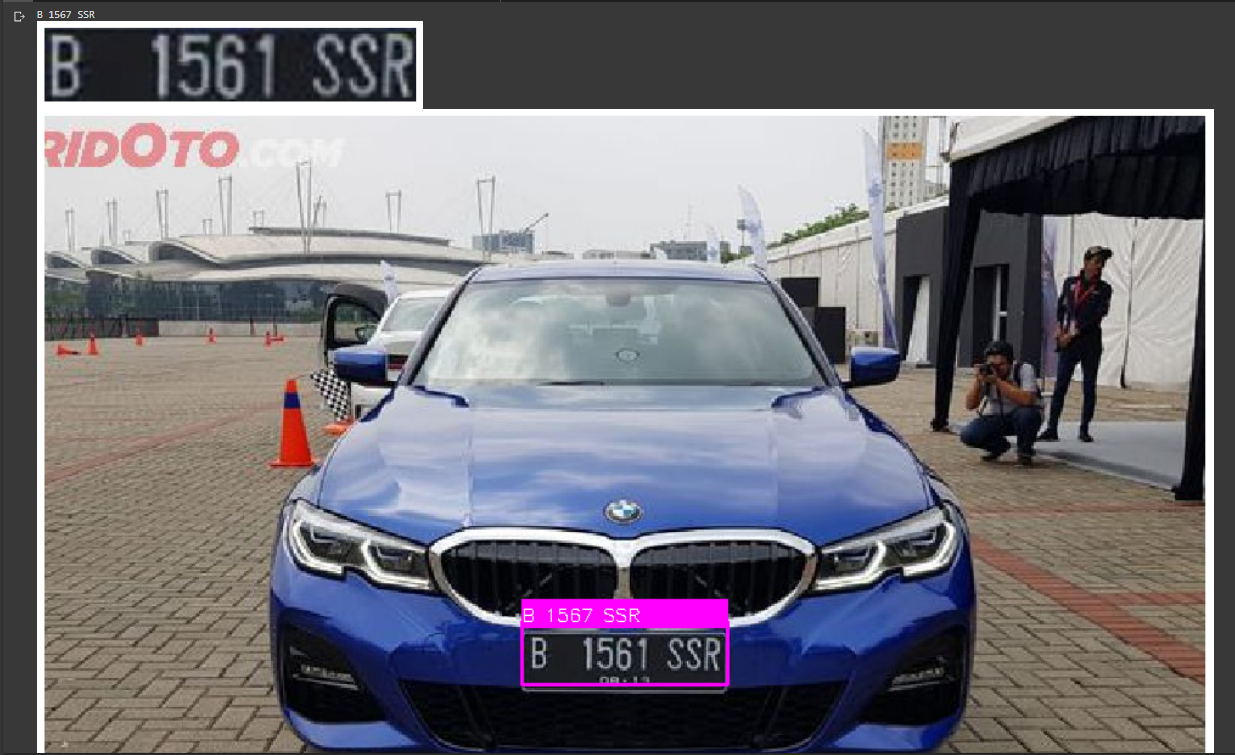
    plt.figure(figsize=(20, 20))

    plt.imshow(img[:, :, ::-1])

    plt.axis('off')

    plt.show()

Hasil nya adalah :



Untuk merubah menjadi grayscale adalah

plt.imshow(thresh, cmap="gray")

plt.axis('off')

plt.show()



Untuk melihat kontur nya dapat menggunakan kode ini

for cnt in contours:

    cv2.drawContours(roi, [cnt], 0, (255, 0, 0), 1)

for pt in pts :

    cv2.circle(roi, (pt[0], pt[1]), 3, (255, 0, 255))

plt.figure(figsize=(20,20))

plt.imshow(roi[:,:,::-1])

plt.axis('off')

plt.show()



Link Google colabs

<https://colab.research.google.com/drive/1_rsiYAIhz1FhhHKF5giq9W5oxPIxP9_y?usp=sharing#scrollTo=j-nolvbW5QRv>