Deteksi Plat Nomor Kendaraan

1120007 - 1121024 - 1121028 - 1121029 - 1121030

Abstrak

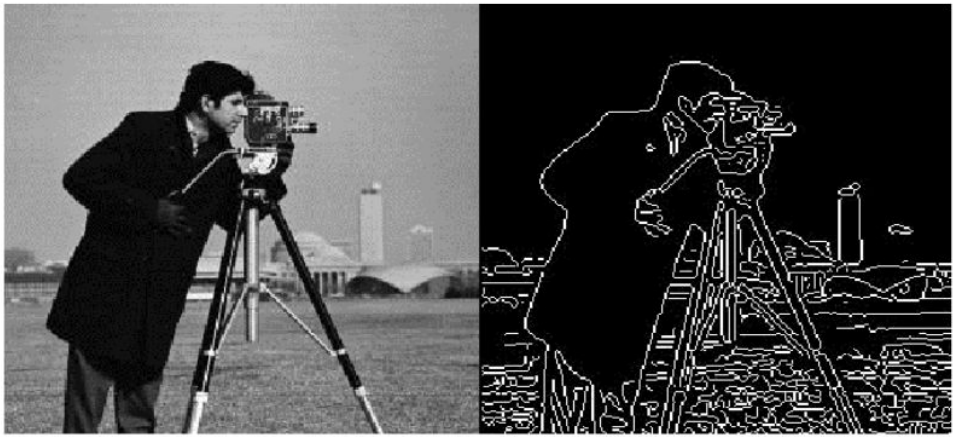
Deteksi plat nomor kendaraan adalah salah satu aplikasi dari pengolahan citra yang digunakan untuk mengidentifikasi plat nomor suatu kendaraan dalam citra.

Metode

Metode yang digunakan untuk deteksi plat nomor kendaraan

Canny Edge Detection Algorithm

Canny Edge Detection Algorithm adalah salah satu algoritma untuk mendeteksi tepi yang paling optimal.



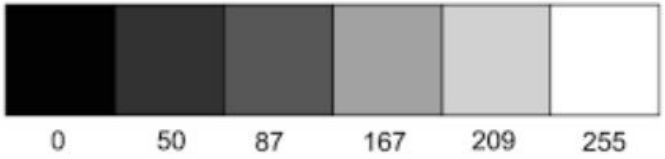
Citra sebelum dan sesudah proses Canny Edge Detection

Gaussian Blur

Gaussian Blur adalah teknik yang menggunakan operasi konvolusi antara citra dan filter kernel.

Grayscale Color

Citra grayscale adalah citra yang hanya memiliki satu tingkat keabuan. Warna abu-abu dari citra grayscale adalah warna kanal R (merah), G (hijau), dan B (biru), yang memiliki intensitas yang sama.



Contoh grayscale pixel value

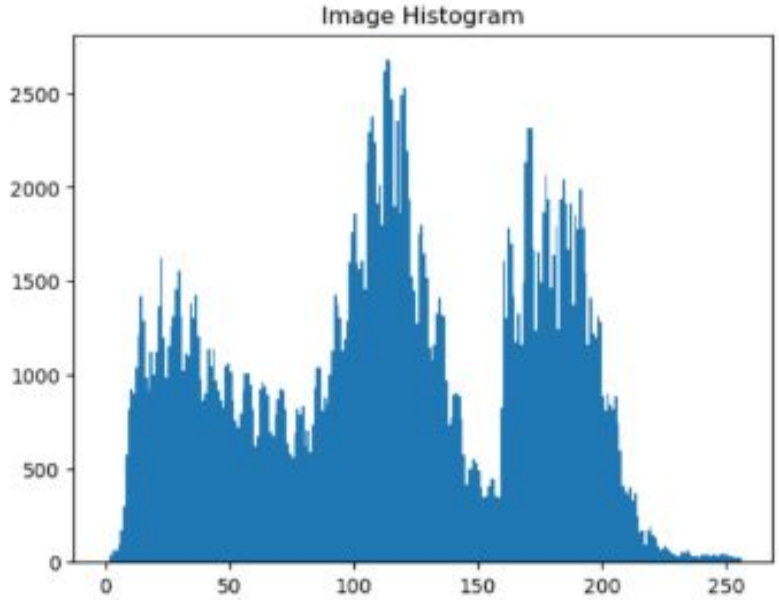
Implementasi

Implementasi pengolahan citra untuk deteksi
plat nomor kendaraan

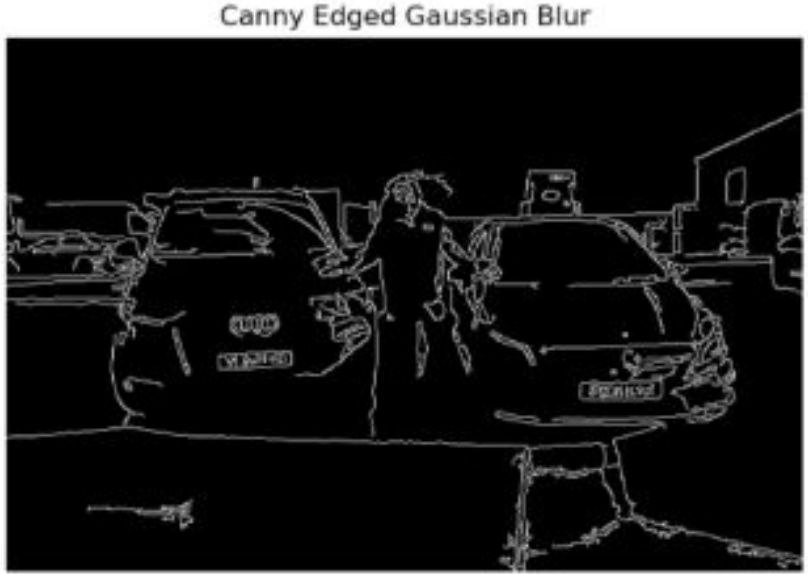
In [2]:
```
img = cv2.imread("1_Shaun-Elliot-from-Hexham-with-the-two-cars-he-won-in-online-draws2.png")

plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.axis("off")
plt.title("Original Image")
plt.show()
```

In [3]:
```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
gaus = cv2.GaussianBlur(gray, (5, 5), 0)

plt.imshow(cv2.cvtColor(gaus, cv2.COLOR_BGR2RGB))
plt.axis("off")
plt.title("Preprocessed Image Gaussian Blur")
plt.show()
```

In [4]:
```
plt.hist(gray.ravel(),256,[0,256])
plt.title('Image Histogram')
plt.show()
```



In [5]:
```
canny_gaus = cv2.Canny(gaus, 30, 200)

plt.imshow(cv2.cvtColor(canny_gaus, cv2.COLOR_BGR2RGB))
plt.axis("off")
plt.title("Canny Edged Gaussian Blur")
plt.show()
```



In [6]:
```
img1 = img.copy()
cnt_canny_gaus, new = cv2.findContours(canny_gaus.copy(), cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)
cv2.drawContours(img1, cnt_canny_gaus, -1, (0, 255, 0), 2)

plt.imshow(cv2.cvtColor(img1, cv2.COLOR_BGR2RGB))
plt.axis("off")
plt.title("Canny Edged Gaussian Blur")
plt.show()
```

In [7]:
```
img1_sorted = img.copy()
cnt_canny_gaus_sorted = sorted(cnt_canny_gaus, key = cv2.contourArea, reverse = True)[:30]
cv2.drawContours(img1_sorted, cnt_canny_gaus_sorted, -1, (0, 255, 0), 2)

plt.imshow(cv2.cvtColor(img1_sorted, cv2.COLOR_BGR2RGB))
plt.axis("off")
plt.title("Canny Edged Gaussian Blur")
plt.show()
```

In [8]:
```
img1_loop_contour = img.copy()

screenCntx = []

for c in cnt_canny_gaus_sorted:
    contour_perimeter = cv2.arcLength(c, True)
    approx = cv2.approxPolyDP(c, 0.018 * contour_perimeter, True)

    if len(approx) == 4:
        screenCntx.append(approx)


cv2.drawContours(img1_loop_contour, screenCntx, -1, (0, 255, 0), 3)

plt.imshow(cv2.cvtColor(img1_loop_contour, cv2.COLOR_BGR2RGB))
plt.axis("off")
plt.title("Canny Edged Gaussian Blur")
plt.show()
```

Analisis Pengujian

Canny Edged Gaussian Blur

Canny Edged Gaussian Blur

Canny Edged Gaussian Blur

Canny Edged Gaussian Blur

Canny Edged Gaussian Blur

Canny Edged Gaussian Blur

Thanks