

7tv3cmn4h

January 30, 2024

Part 1 - B: I have tried to explain what my approach is and the things i tried for the Part1-B question - correctly identifying the number of fingers in the frame

I captured an image of the finger so that i can find a solution to the picture first and then apply the same to the frames

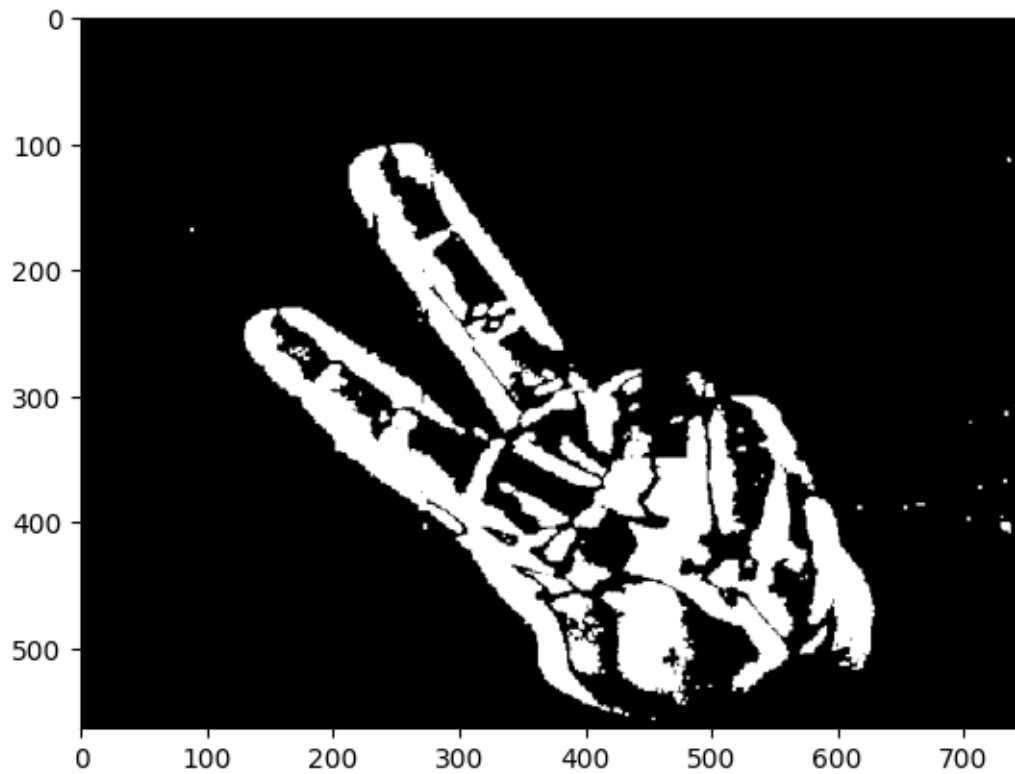
```
[9]: import cv2
import matplotlib.pyplot as plt
#you can use pillow
fname = 'finger.png'
# load an RGB image
data = cv2.imread(fname)
red = data[..., 0] # data[:, :, 0]
green = data[..., 1]
blue = data[..., 2]
# Luminosity
finger = 0.21 * red + 0.72 * green + 0.07 * blue #numbers based on visual
↪science
```

```
[20]: #convert al the pixel values greater than 0 to 255
import numpy as np
binarized_image = np.where(finger > 0, 255, 0).astype(np.uint8)
final_image = binarized_image.copy()
np.unique(final_image)
```

```
[20]: array([ 0, 255], dtype=uint8)
```

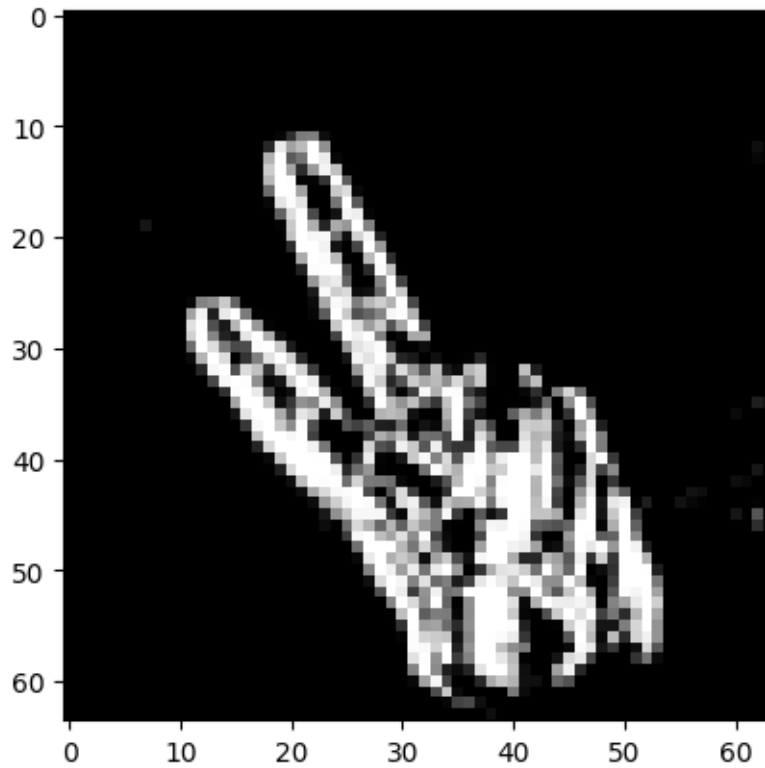
```
[21]: plt.imshow(final_image,cmap = "gray")
```

```
[21]: <matplotlib.image.AxesImage at 0x26c3406b108>
```



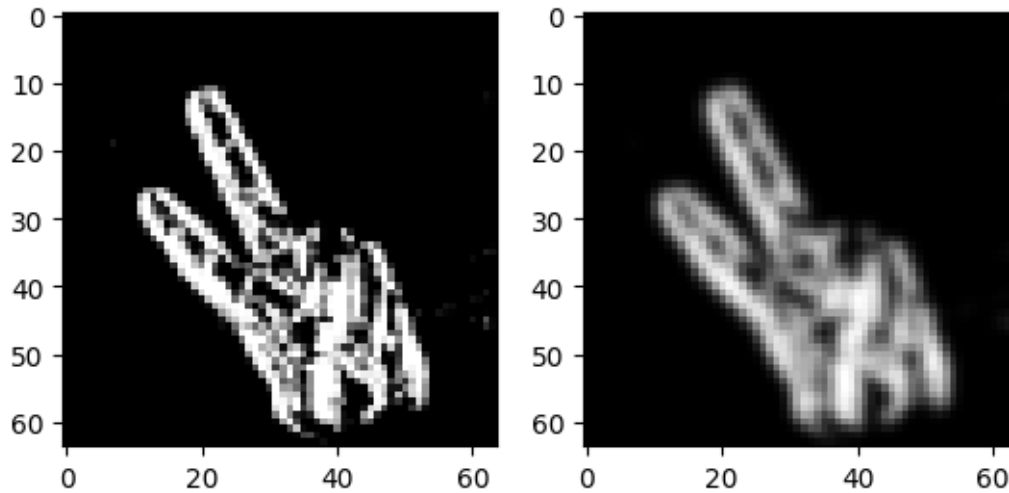
```
[22]: # Resize the image to 64 by 64 pixels using cv2
      resized_image = cv2.resize(final_image, (64, 64), interpolation=cv2.INTER_AREA)
      plt.imshow(resized_image)
```

```
[22]: <matplotlib.image.AxesImage at 0x26c33ae48>
```



I have applied gaussian filter so that i can smoothen the edges

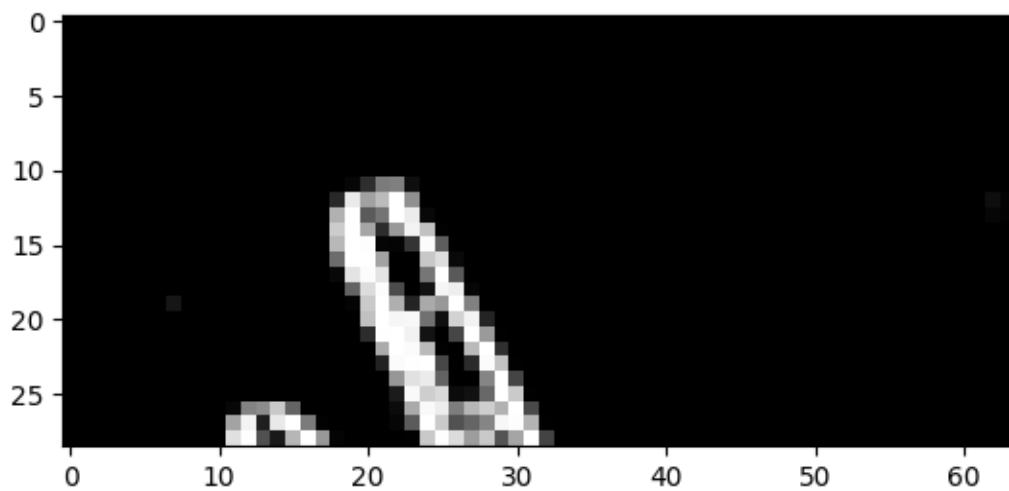
```
[24]: from scipy.ndimage import gaussian_filter
fig = plt.figure()
plt.gray() # show the filtered result in grayscale
ax1 = fig.add_subplot(121) # left side
ax2 = fig.add_subplot(122) # right side
result = gaussian_filter(resized_image, sigma=1)
ax1.imshow(resized_image)
ax2.imshow(result)
plt.show()
```



```
[47]: for row_index, row in enumerate(resized_image):
      # Check if any value in the row is greater than 30
      if any(value > 0 for value in row): #threshold is 20
          split_index = row_index
          print(f'Row {row_index} contains values greater than 20')
          break
      split_index = row_index + 18
      plt.imshow(resized_image[:split_index])
```

Row 11 contains values greater than 20

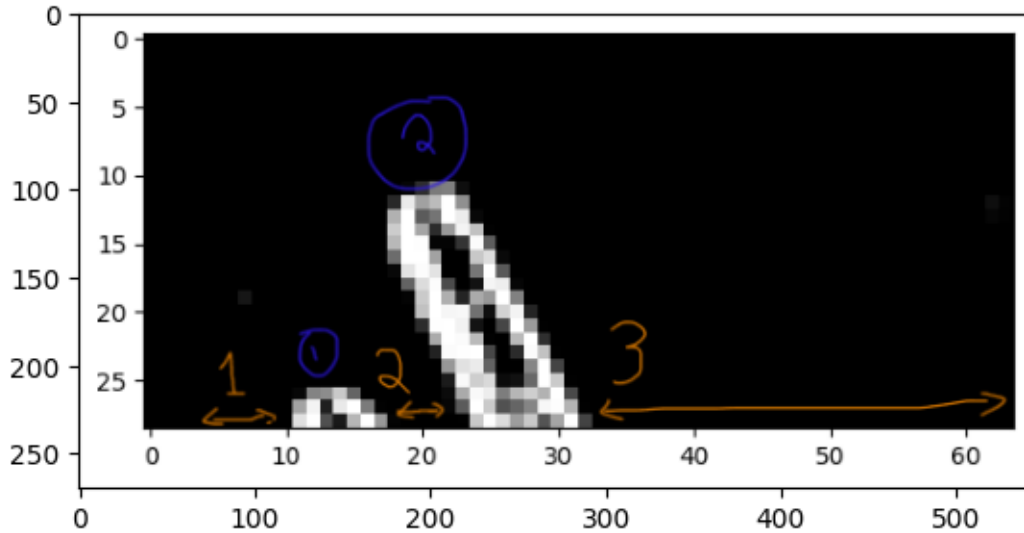
```
[47]: <matplotlib.image.AxesImage at 0x26c3e6dae48>
```



Once the image is split, get the (split\_index + 20)th row assuming all fingers would be part of this row

```
[62]: image= cv2.imread("output.png")
      plt.imshow(image)
```

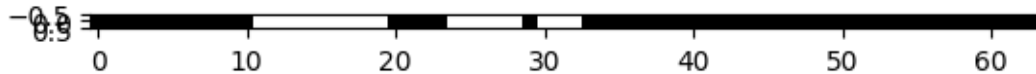
```
[62]: <matplotlib.image.AxesImage at 0x26c33164a88>
```



```
[48]: finger_image_row = np.where(resized_image[split_index:split_index+1] > 0, 255, 0).astype(np.uint8)
```

```
[52]: finger_row = resized_image[split_index-1].tolist()
      plt.imshow(finger_image_row)
```

```
[52]: <matplotlib.image.AxesImage at 0x26c3e82c148>
```



and then count the number of times white gap appears in a black row.

```
[57]: count = 0
      for i in range(len(finger_row)):
          if finger_row[i] < 1:
              continue
          if finger_row[i] > 0 :
```

```
pass
if finger_row[i+1] < 1 :
    count = count + 1
else:
    continue
print("The number of fingers are",count)
```

The number of fingers are 2