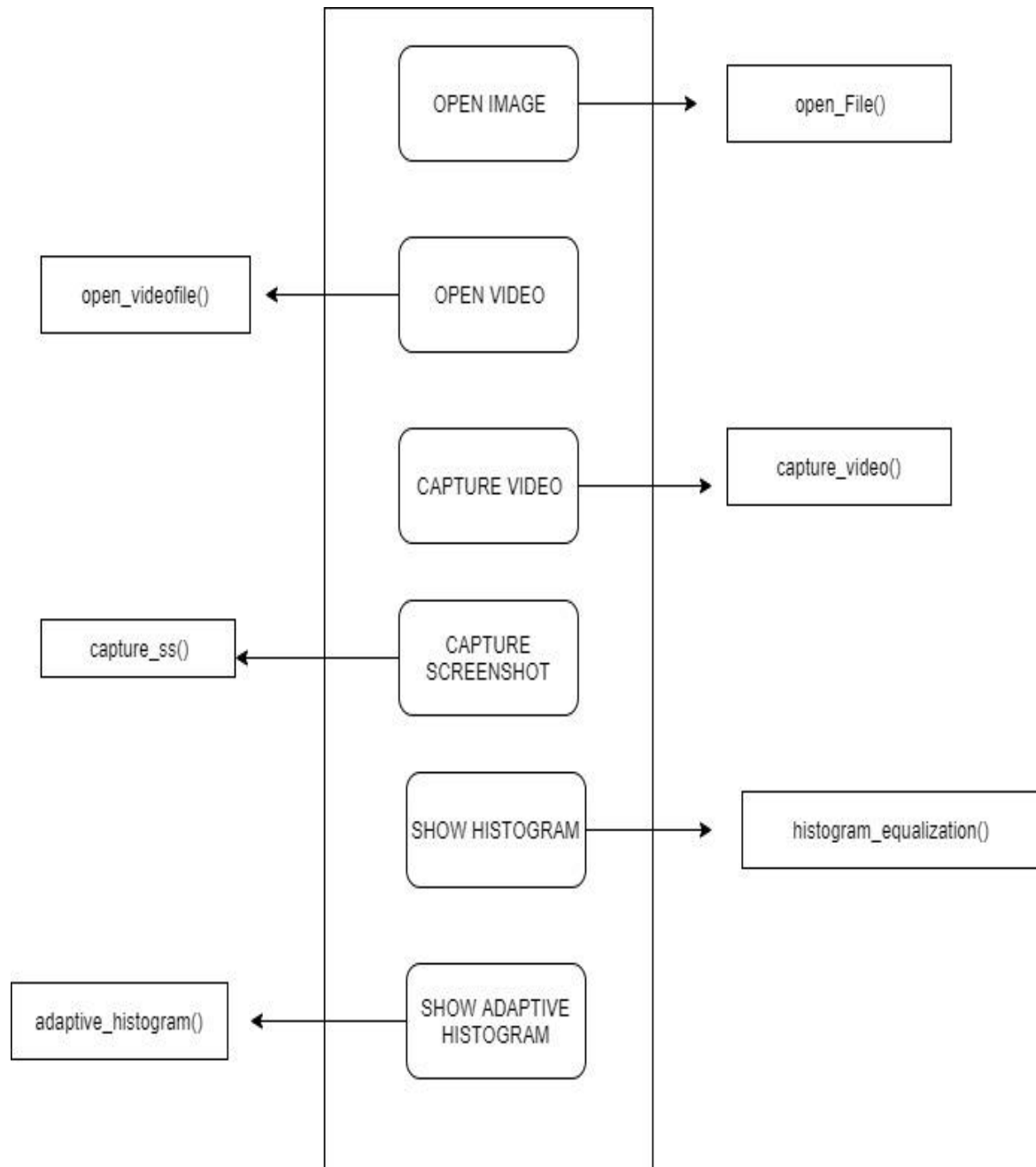


Name: Birva Patel

Student-id: 1111092

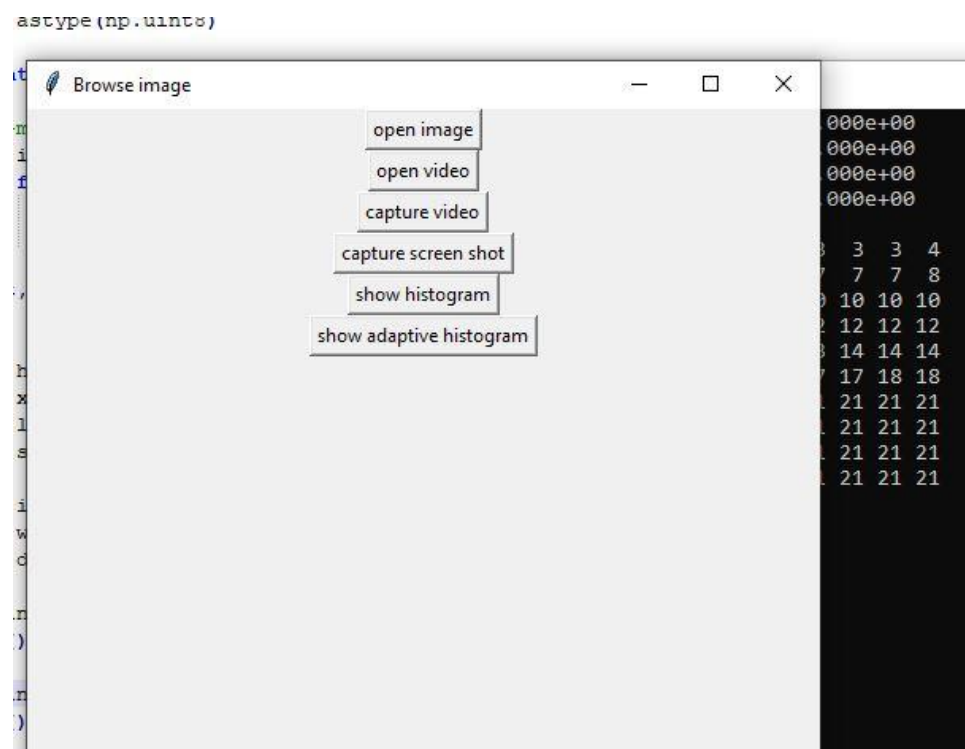
DESIGN DOCUMENT

GUI LAYOUT:



GUI Implementation:

1. Open Image
2. Open Video
3. Capture Video
4. Capture Screenshot
5. Show Histogram
6. Show adaptive histogram



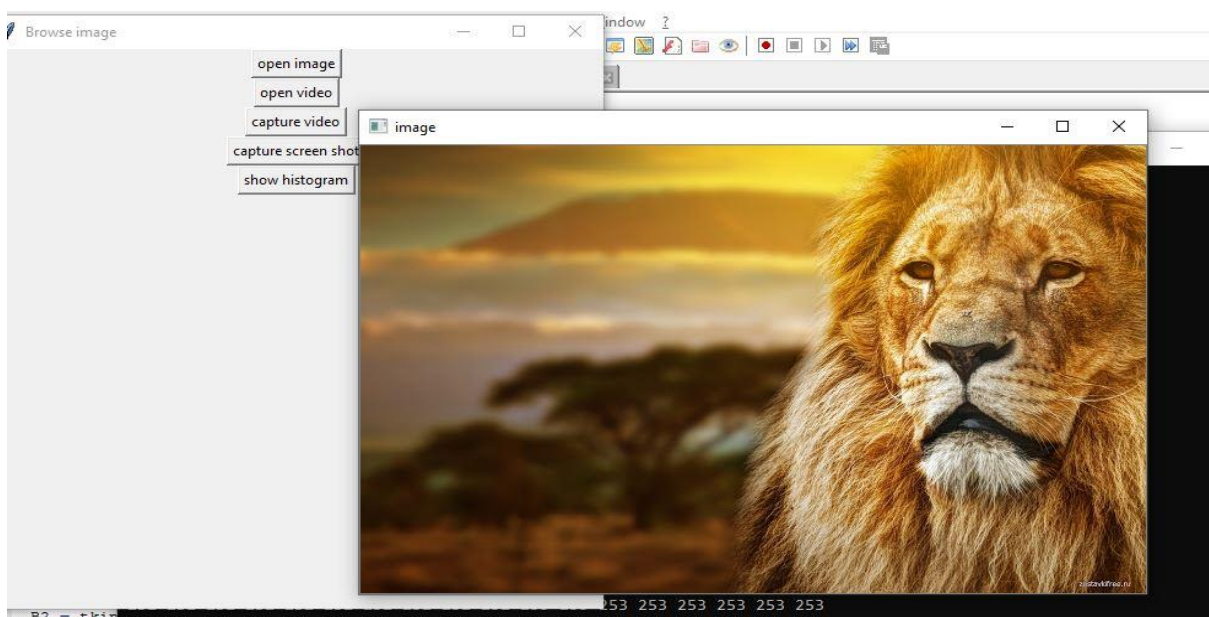
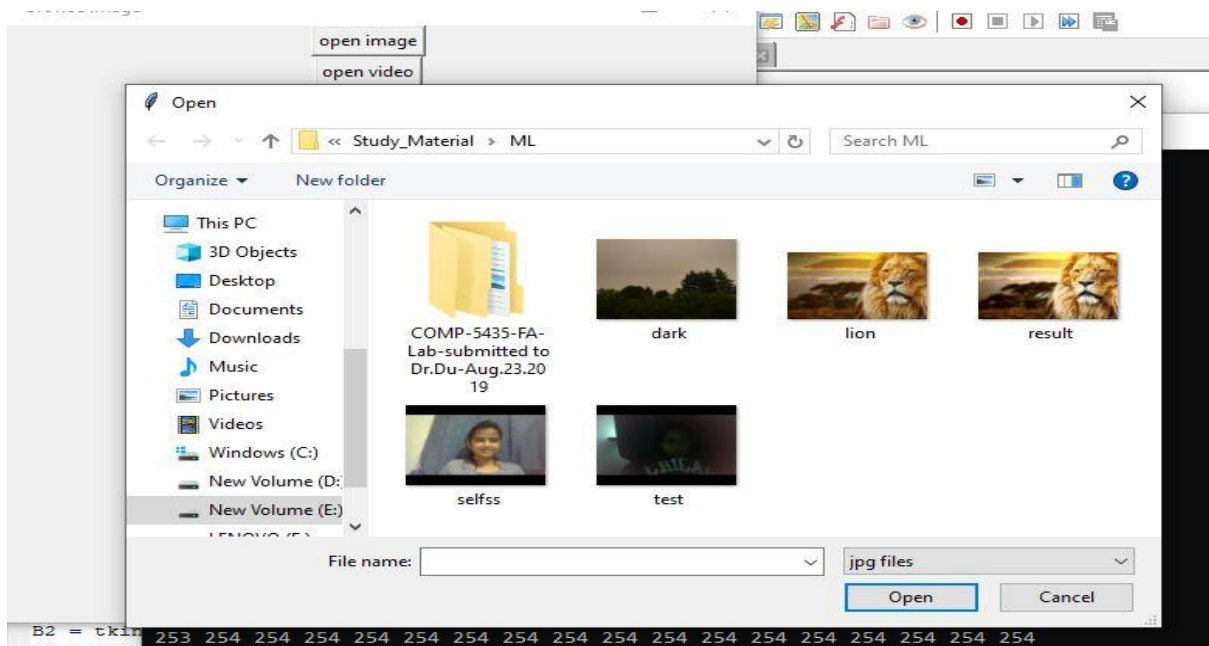
OPEN IMAGE:

I have created a button using tkinter to browse the image from file. On click it will open the file explorer and from that we can see image on screen. One functionality is that you can only access the images, other formats are unsupported.

For that I have used `askopenfilename()` function which will open directory on click of open image button. Using `imshow()` function, we can display the selected image.

```
def open_File():

    rw.filename = askopenfilename(filetypes=[("images","*.png"),("jpg files","*.jpg")])
    img = cv2.imread(rw.filename)
    cv2.imshow('image',img)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
B1 = tkinter.Button(rw, text ="open image", command = open_File)
B1.pack()
```



OPEN VIDEO:

I have created a button using tkinter to browse the video from file. On click it will open the file explorer and from that we can watch video on screen. One functionality is that you can only access the video, other formats are unsupported.

For that I have used askopenfilename() function which will open directory on click of open video button. Using VideoCapture() and cvtColor() function, we can display the selected video.

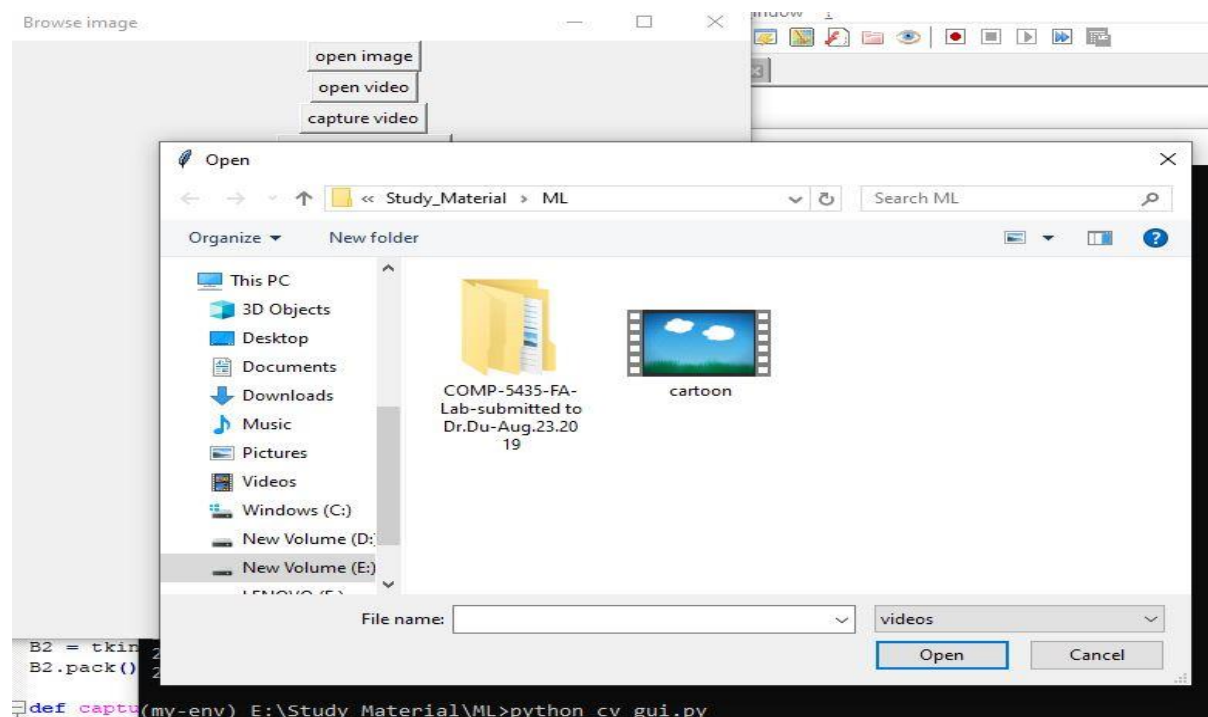
```
def open_videofile():
    rw.filename = askopenfilename(filetypes=[("videos", "*.mp4"), ("avi files", "*.avi")])
    cap = cv2.VideoCapture(rw.filename, 0)

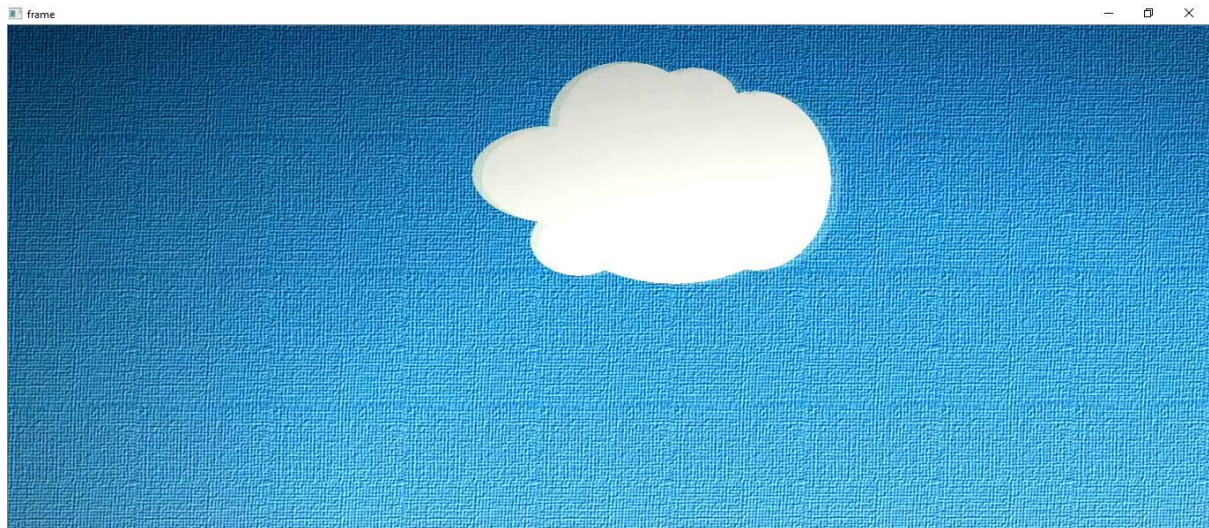
    while(cap.isOpened()):
        ret, frame = cap.read()

        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2BGRA)

        cv2.imshow('frame', gray)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()
B2 = tkinter.Button(rw, text = "open video", command = open_videofile )
B2.pack()
```





CAPTURE VIDEO:

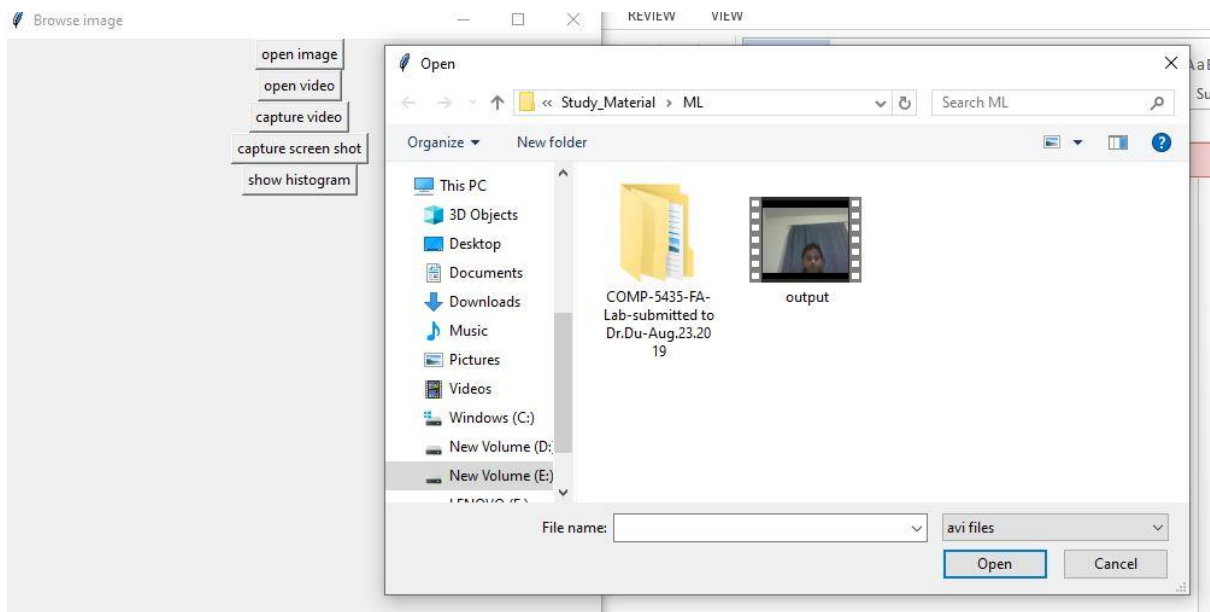
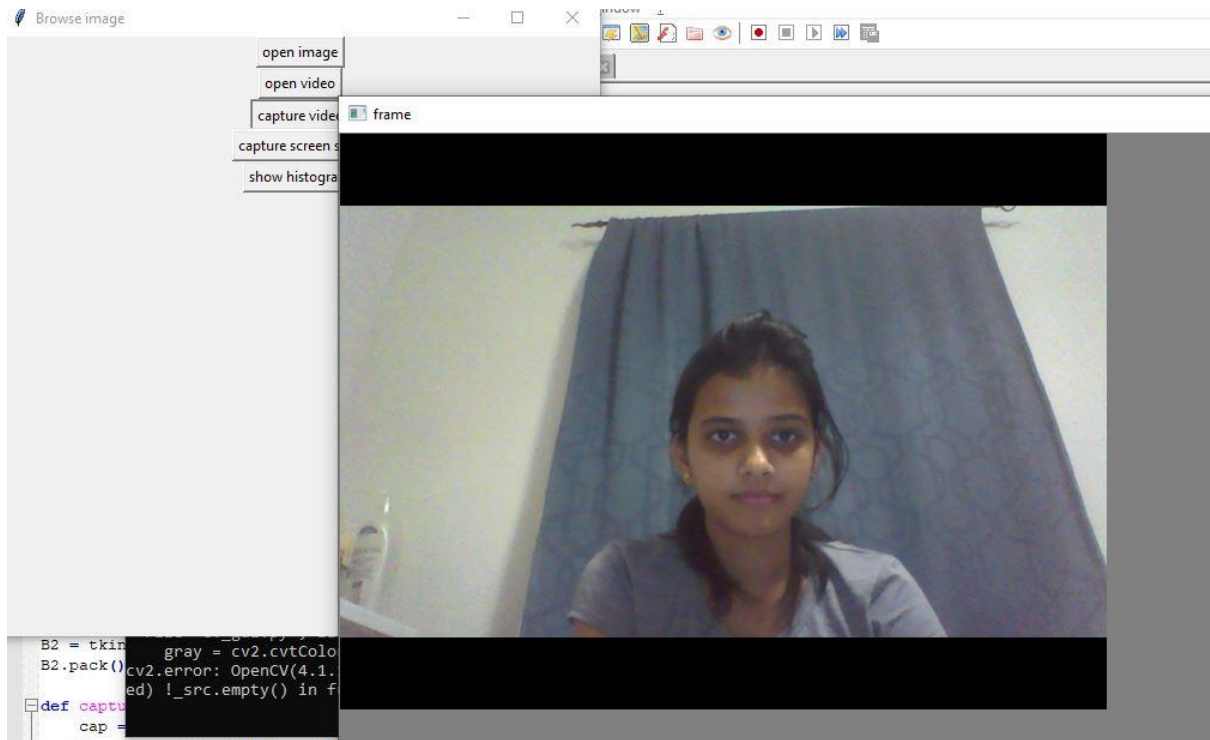
I have created a button using tkinter to start the webcam and record the video then by clicking q it will stop recording. Also that video will be saved to the file directory..We can access that video by open video button.

For that I have used VideoCapture() and VideoWriter() functions which will capture the video and store the video. In this process we will operate the function frame by frame. Third screenshot shows the saved video in file named as output.

```
def capture_video():
    cap = cv2.VideoCapture(0)
    fourcc = cv2.VideoWriter_fourcc(*'XVID')
    out = cv2.VideoWriter('output.avi',fourcc, 20.0, (640,480))

    while(True):
        ret, frame = cap.read()    # reading frame-by-frame
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2BGRA)    # actual operation on frame
        out.write(frame)
        cv2.imshow('frame',gray)    # Display the output frame
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    # release the capture
    cap.release()
    out.release()
    cv2.destroyAllWindows()
B3 = tkinter.Button(rw, text ="capture video", command = capture_video )
B3.pack()
```

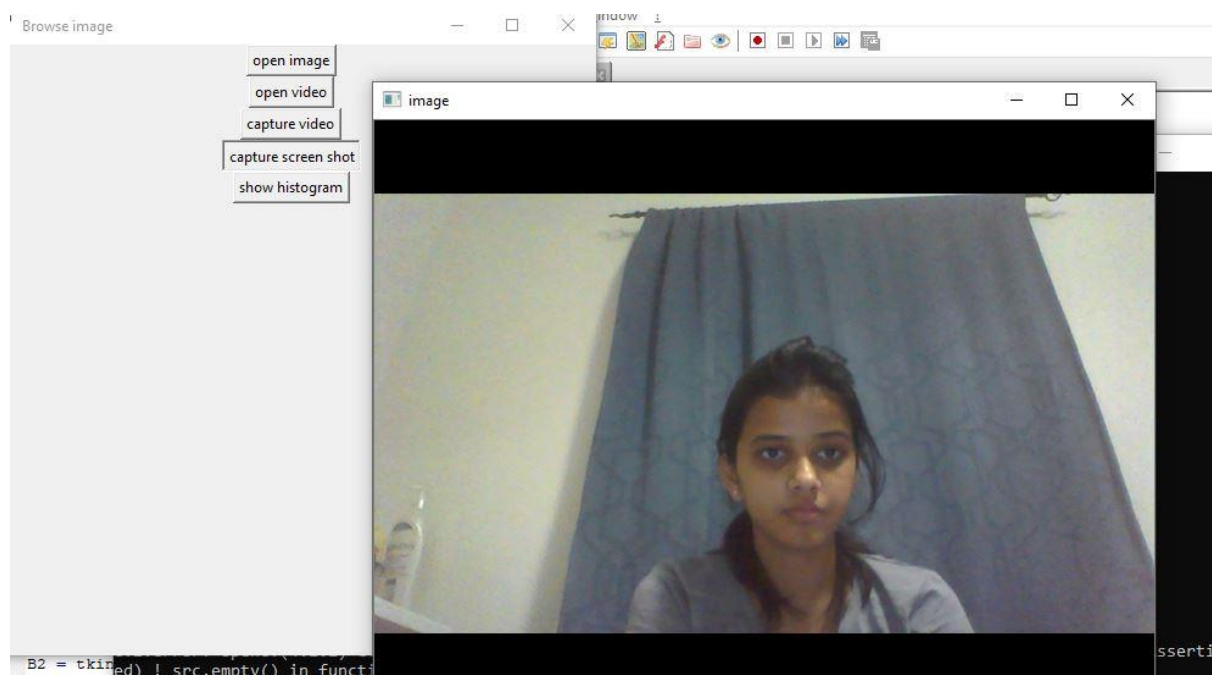


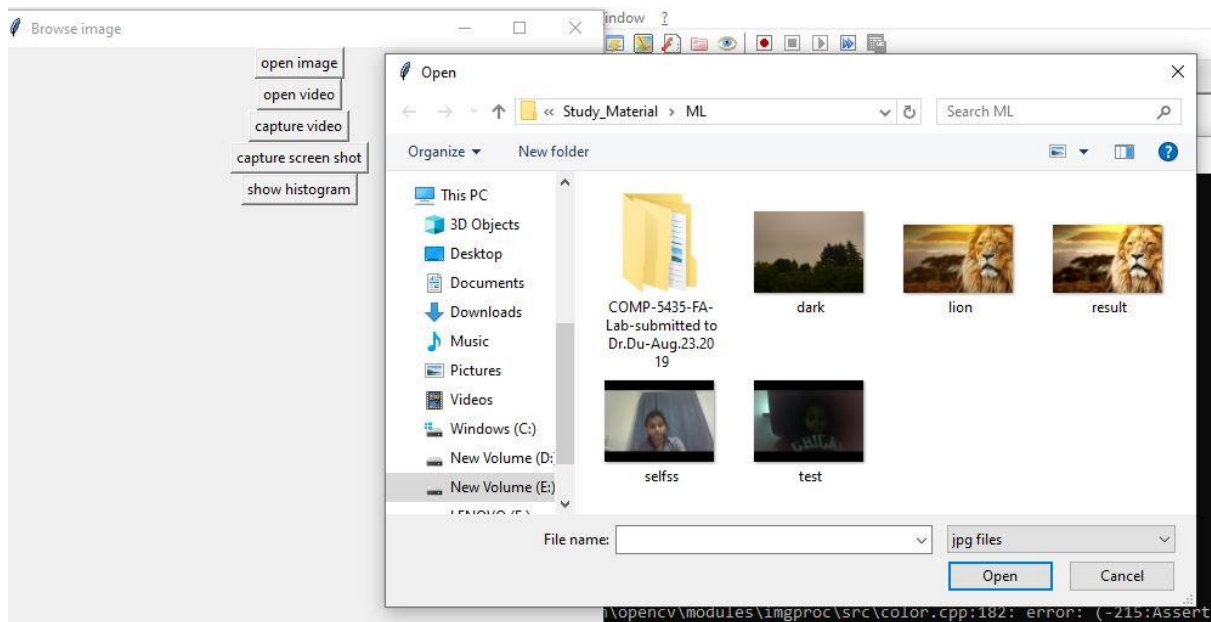
CAPTURE SCREEN SHOT:

I have created a button using tkinter to start the webcam and by clicking s it will capture the screenshot and save it in file. We can access that screenshot by open image button mentioned above.

For that I have used VideoCapture() function. Which will capture the image and image will be saved in respective directory. The last screen shot shows the saved image named as selfss.

```
def capture_ss():  
    camera = cv2.VideoCapture(0)  
    while True:  
        return_value, image = camera.read()  
        gray = cv2.cvtColor(image, cv2.COLOR_BGR2BGRA)  
  
        cv2.imshow('image', gray)  
        if cv2.waitKey(1) & 0xFF == ord('s'):  
            cv2.imwrite('selfss.jpg', image)  
            break  
    camera.release()  
    cv2.destroyAllWindows()  
B4 = tkinter.Button(rw, text = "capture screen shot", command = capture_ss )  
B4.pack()
```





SHOW HISTOGRAM:

I have created a button using tkinter to load the image from directory and after that it will process on image to make it contrast image than before (clearer than before). Also the graph will be displayed to show hist values.

For that I have loaded an color image in grayscale. The original image will be displayed before equalization. by performing histogram equalization, using for loop, we will store the histogram in b variable. then it will Re-map values from equalized histogram into the image.


```

def histogram_equalization():
    rw.filename = askopenfilename(filetypes=[("image*", "*.png"), ("jpg file*", "*.jpg")])
    # load an color image in grayscale
    img = cv2.imread(rw.filename, 0)

    #To display image before equalization
    cv2.imshow('image', img)
    cv2.waitKey(0)

    a = np.zeros((256, ), dtype=np.float16)
    b = np.zeros((256, ), dtype=np.float16)

    height, width = img.shape

    #finding histogram
    for i in range(width):
        for j in range(height):
            q = img[j, i]
            a[q] = a[q] + 1

    print(a)

    #performing histogram equalization
    tmp = 1.0 / (height * width)
    b = np.zeros((256, ), dtype=np.float16)

    for i in range(256):
        for j in range(i + 1):
            b[i] += a[j] * tmp:
        b[i] = round(b[i] * 255) }

    # b now contains the equalized histogram
    b = b.astype(np.uint8)

    print(b)

    #Re-map values from equalized histogram into the image
    for i in range(width):
        for j in range(height):
            q = img[j, i]
            img[j, i] = b[q]

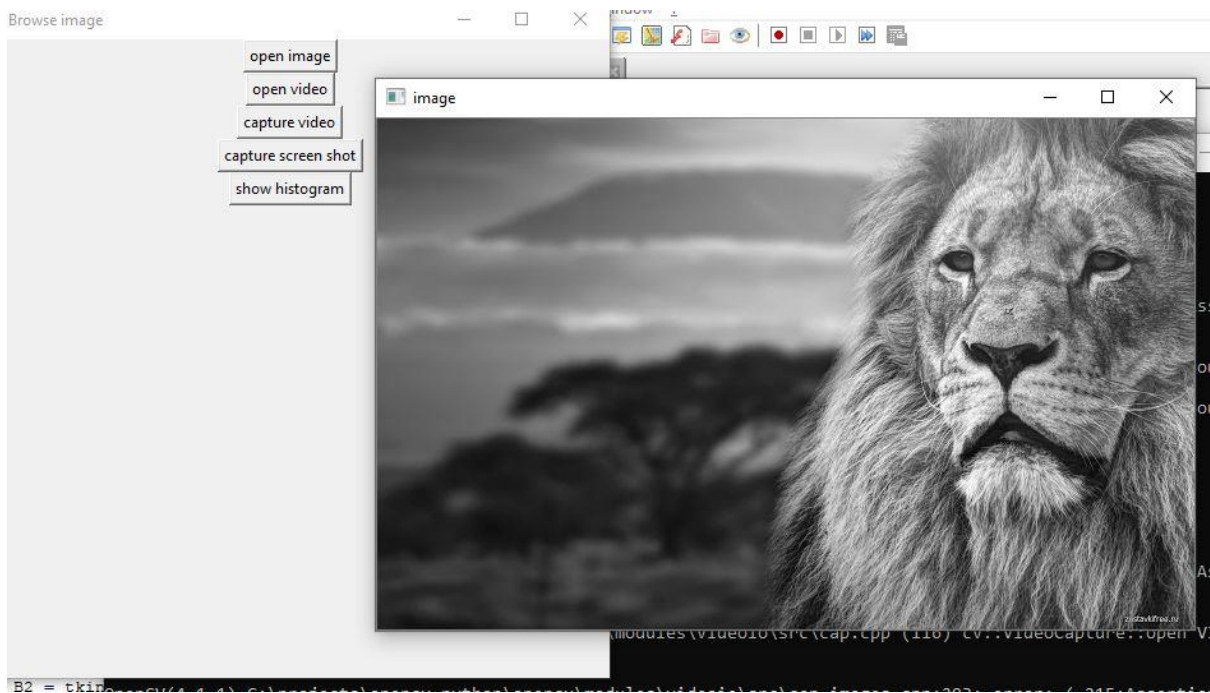
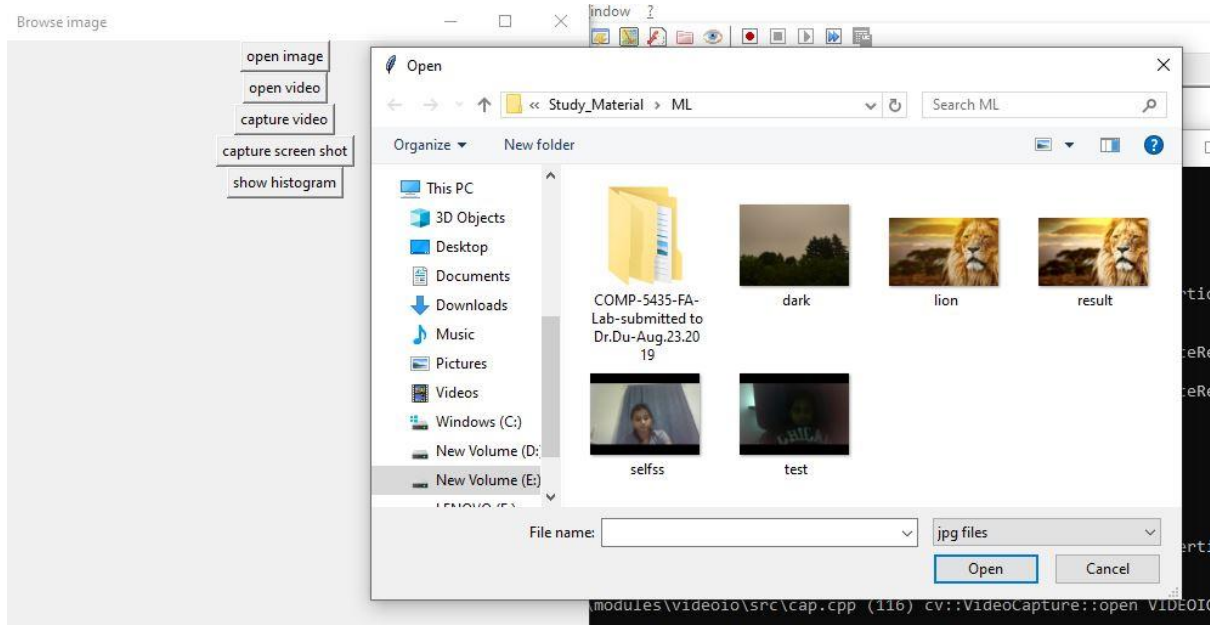
    hist, bins = np.histogram(img.flatten(), 256, [0, 256])

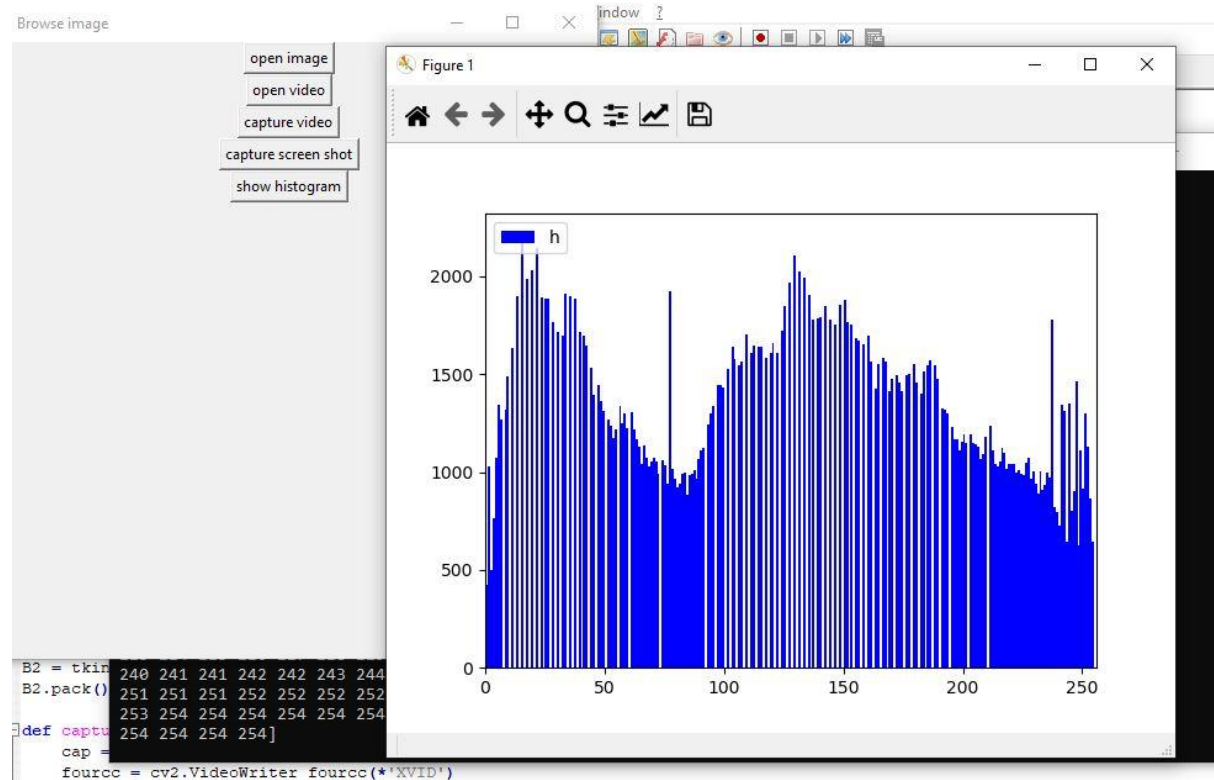
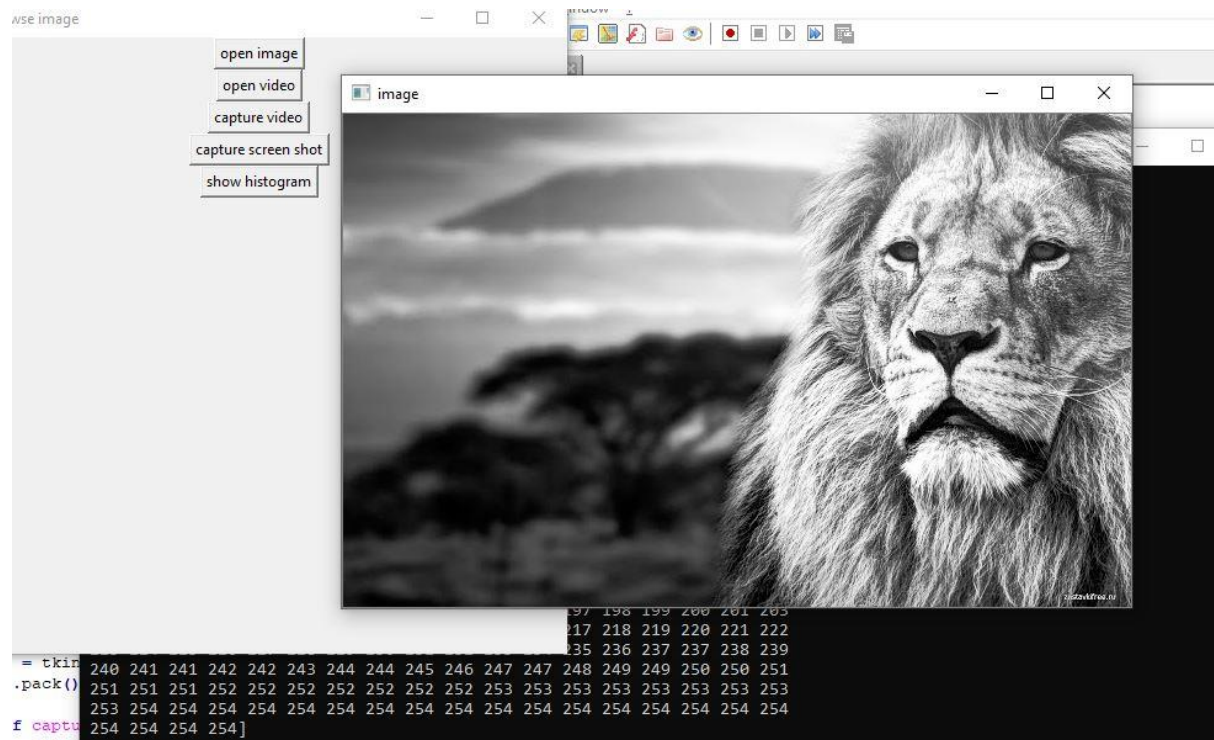
    plt.hist(img.flatten(), 256, [0, 256], color = 'b')
    plt.xlim([0, 256])
    plt.legend(['histogram'], loc = 'upper left')
    plt.show()

    cv2.imshow('image', img)
    cv2.waitKey(0)
    cv2.destroyAllWindows()

B5 = tkinter.Button(rw, text = "show histogram", command = histogram_equalization )
B5.pack()

```





SHOW ADAPTIVE HISTOGRAM:

I have created a button using tkinter to load the image from directory and after that it will measure the shape and size of image. then by dividing the image in frames it will contrast the image. We will use array to store the divided frames.

```
img = cv2.imread('lion.jpg',0)
print (img)
img_size=img.shape
print (img_size)

img_mod = np.zeros((600, 800))

for i in range(0,img_size[0]-30):
    for j in range(0,img_size[1]-30):
        kernel = img[i:i+30,j:j+30]
        for k in range(0,30):
            for l in range(0,30):
                element = kernel[k,l]
                rank = 0
                for m in range(0,30):
                    for n in range(0,30):
                        if(kernel[k,l]>kernel[m,n]):
                            rank = rank + 1
                img_mod[i,j] = ((rank * 255 )/900)

im = np.array(img_mod, dtype = np.uint8)
cv2.imwrite('target.jpg',im)
```

