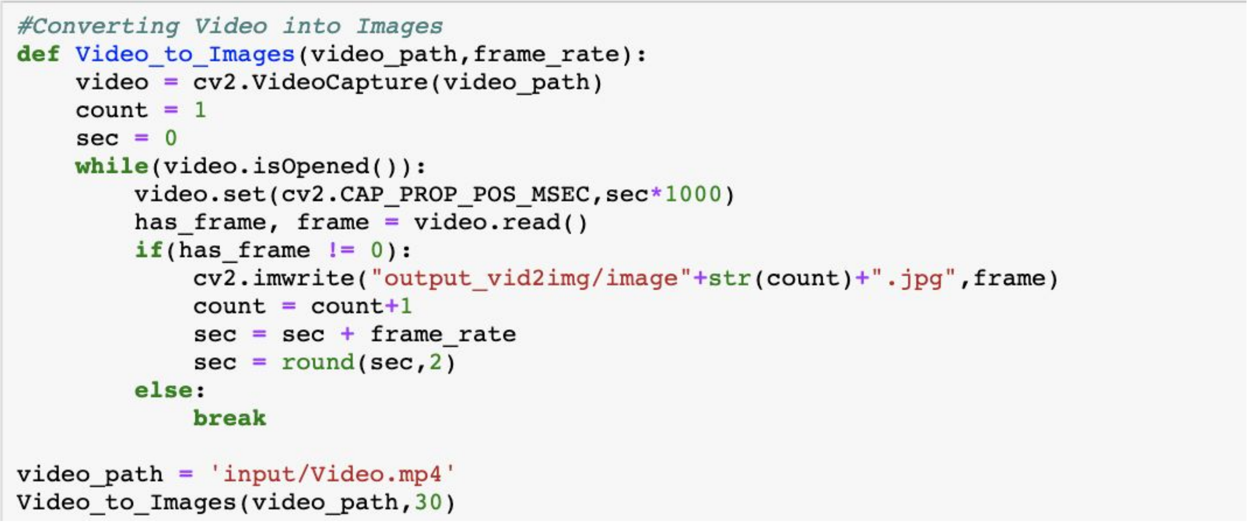
Computer Vision Assignment 0

Gowri Lekshmy
20171053

Question 1

Video to Images

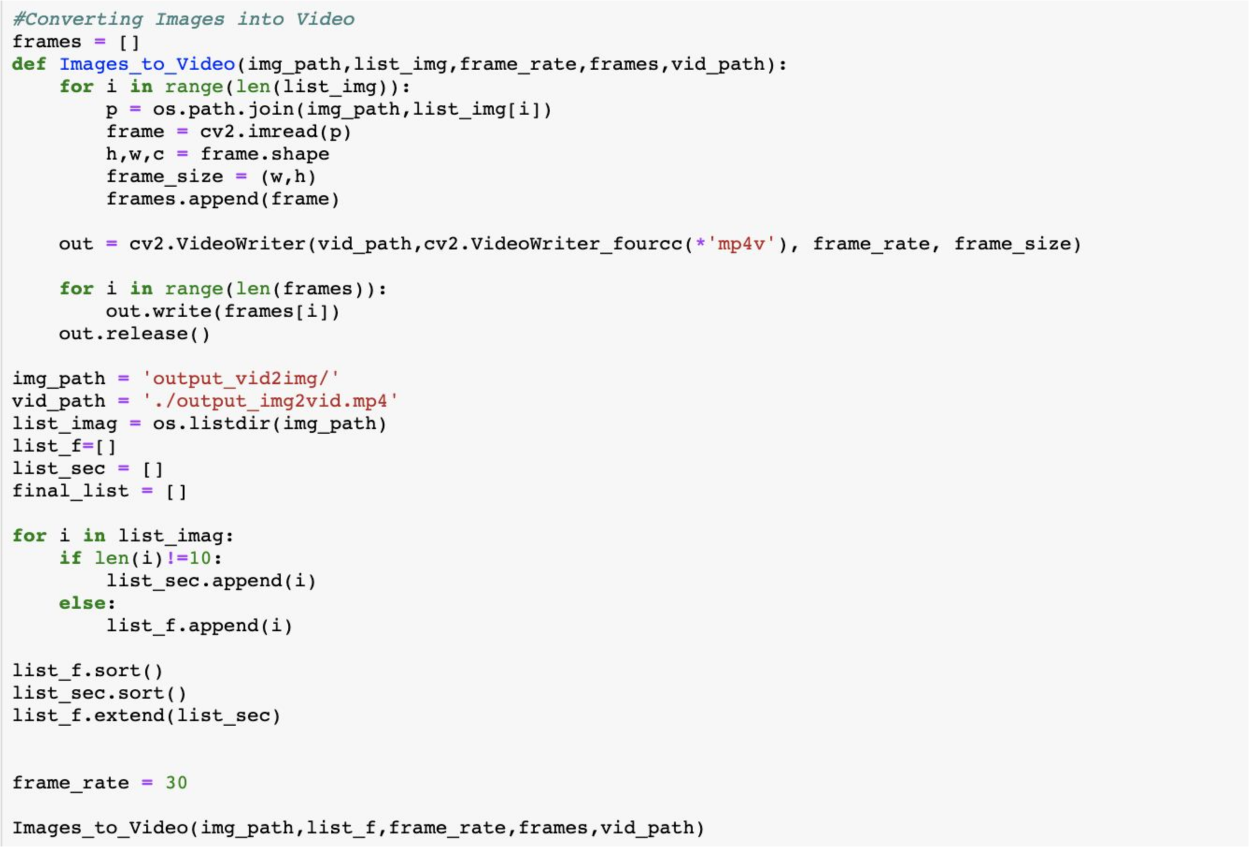
An input video is taken and the user can specify a frame rate. Here, I have specified a frame rate of 30. The input video is converted into images at an interval of 30 seconds and these images are stored in a folder output_vid2img. This is done by checking whether the video has a frame or not until the end of the video and then it is saved as an image according to the frame rate.



```
#Converting Video into Images
def Video_to_Images(video_path,frame_rate):
    video = cv2.VideoCapture(video_path)
    count = 1
    sec = 0
    while(video.isOpened()):
        video.set(cv2.CAP_PROP_POS_MSEC,sec*1000)
        has_frame, frame = video.read()
        if(has_frame != 0):
            cv2.imwrite("output_vid2img/image"+str(count)+".jpg",frame)
            count = count+1
            sec = sec + frame_rate
            sec = round(sec,2)
        else:
            break

video_path = 'input/Video.mp4'
Video_to_Images(video_path,30)
```

Images to Video

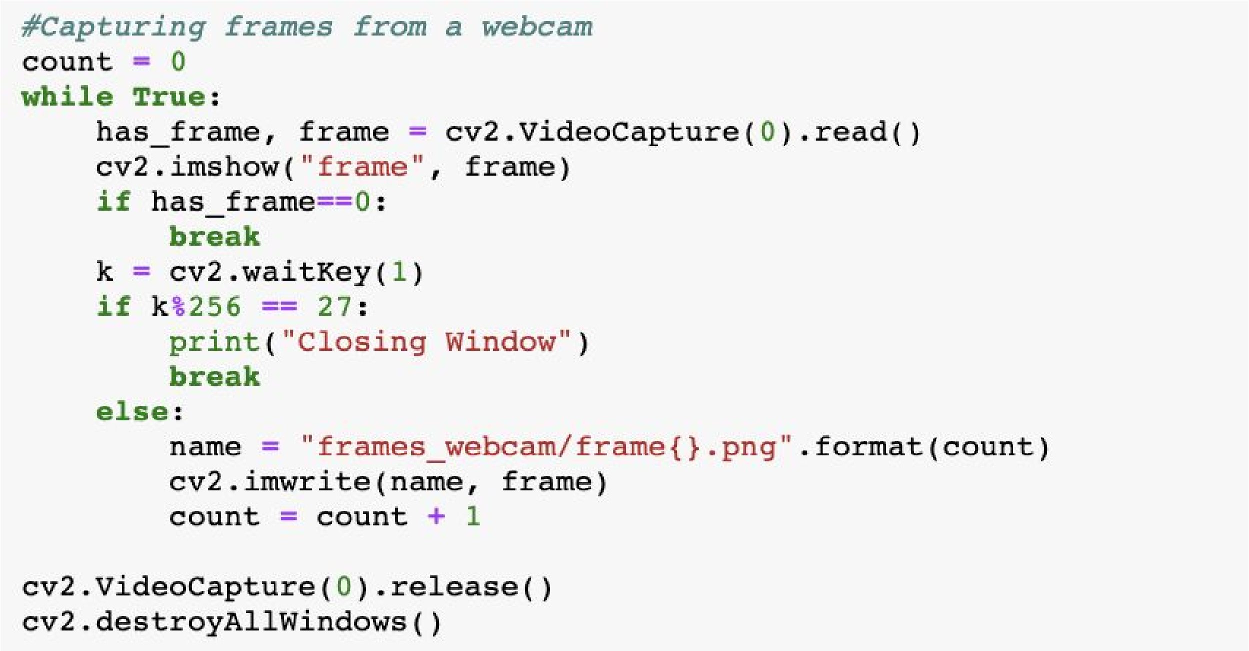
A folder of images is specified by the user and these images are concatenated at a particular frame rate to create a video. The input folder is output_vid2img and the output video is output_img2vid.mp4. The frame rate at which the images are concatenated is 30 sec.



```
#Converting Images into Video
frames = []
def Images_to_Video(img_path,list_img,frame_rate,frames,vid_path):
    for i in range(len(list_img)):
        p = os.path.join(img_path,list_img[i])
        frame = cv2.imread(p)
        h,w,c = frame.shape
        frame_size = (w,h)
        frames.append(frame)

    out = cv2.VideoWriter(vid_path,cv2.VideoWriter_fourcc(*'mp4v'), frame_rate, frame_size)

    for i in range(len(frames)):
        out.write(frames[i])
    out.release()

img_path = 'output_vid2img/'
vid_path = './output_img2vid.mp4'
list_imag = os.listdir(img_path)
list_f=[]
list_sec = []
final_list = []

for i in list_imag:
    if len(i)!=10:
        list_sec.append(i)
    else:
        list_f.append(i)

list_f.sort()
list_sec.sort()
list_f.extend(list_sec)


frame_rate = 30

Images_to_Video(img_path,list_f,frame_rate,frames,vid_path)
```

Question 2

Frames are being captured from a webcam and then stored with an incremental count corresponding to the frame number. To pause the webcam from capturing more frames, the ESC key is used which waits until 1 ms as cv2.waitKey(1) is used.

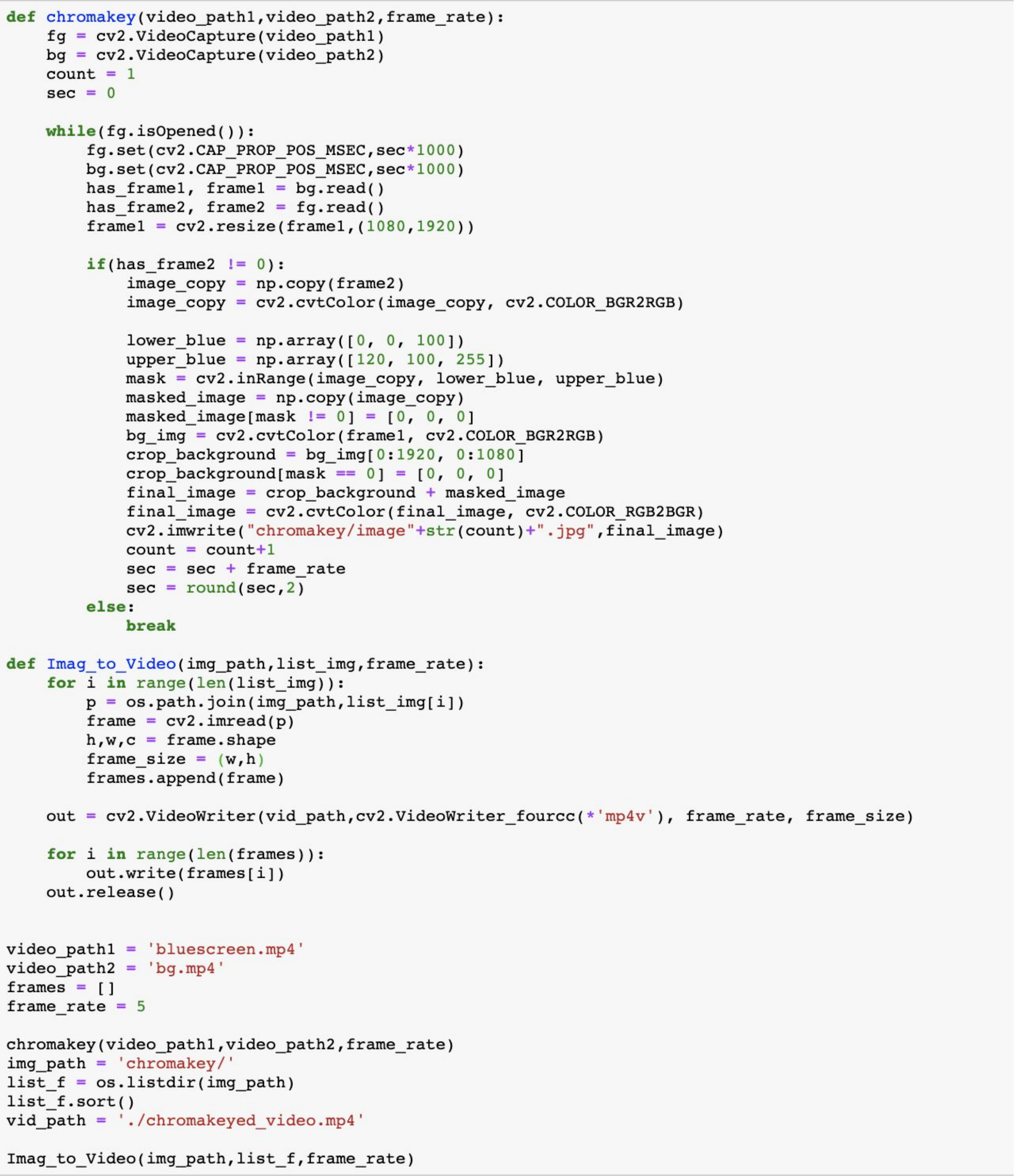


```
#Capturing frames from a webcam
count = 0
while True:
    has_frame, frame = cv2.VideoCapture(0).read()
    cv2.imshow("frame", frame)
    if has_frame==0:
        break
    k = cv2.waitKey(1)
    if k%256 == 27:
        print("Closing Window")
        break
    else:
        name = "frames_webcam/frame{}.png".format(count)
        cv2.imwrite(name, frame)
        count = count + 1

cv2.VideoCapture(0).release()
cv2.destroyAllWindows()
```

Question 3

Chroma Keying for two videos is done by taking one video with myself for the foreground and another video for getting the background.

We try to replace the blue background pixels in the video with me, with the pixels in the background video. This is done by identifying the blue pixels within a particular range and then creating a mask with the foreground pixels as black pixels and others with white pixels. Then the cropped background image and the masked image are added.



```
def chromakey(video_path1,video_path2,frame_rate):
    fg = cv2.VideoCapture(video_path1)
    bg = cv2.VideoCapture(video_path2)
    count = 1
    sec = 0

    while(fg.isOpened()):
        fg.set(cv2.CAP_PROP_POS_MSEC,sec*1000)
        bg.set(cv2.CAP_PROP_POS_MSEC,sec*1000)
        has_frame1, frame1 = bg.read()
        has_frame2, frame2 = fg.read()
        frame1 = cv2.resize(frame1,(1080,1920))

        if(has_frame2 != 0):
            image_copy = np.copy(frame2)
            image_copy = cv2.cvtColor(image_copy, cv2.COLOR_BGR2RGB)

            lower_blue = np.array([0, 0, 100])
            upper_blue = np.array([120, 100, 255])
            mask = cv2.inRange(image_copy, lower_blue, upper_blue)
            masked_image = np.copy(image_copy)
            masked_image[mask != 0] = [0, 0, 0]
            bg_img = cv2.cvtColor(frame1, cv2.COLOR_BGR2RGB)
            crop_background = bg_img[0:1920, 0:1080]
            crop_background[mask == 0] = [0, 0, 0]
            final_image = crop_background + masked_image
            final_image = cv2.cvtColor(final_image, cv2.COLOR_RGB2BGR)
            cv2.imwrite("chromakey/image"+str(count)+".jpg",final_image)
            count = count+1
            sec = sec + frame_rate
            sec = round(sec,2)
        else:
            break

def Imag_to_Video(img_path,list_img,frame_rate):
    for i in range(len(list_img)):
        p = os.path.join(img_path,list_img[i])
        frame = cv2.imread(p)
        h,w,c = frame.shape
        frame_size = (w,h)
        frames.append(frame)

    out = cv2.VideoWriter(vid_path,cv2.VideoWriter_fourcc(*'mp4v'), frame_rate, frame_size)

    for i in range(len(frames)):
        out.write(frames[i])
    out.release()


video_path1 = 'bluescreen.mp4'
video_path2 = 'bg.mp4'
frames = []
frame_rate = 5

chromakey(video_path1,video_path2,frame_rate)
img_path = 'chromakey/'
list_f = os.listdir(img_path)
list_f.sort()
vid_path = './chromakeyed_video.mp4'

Imag_to_Video(img_path,list_f,frame_rate)
```

Challenges faced:

1. It is difficult to adjust the frame rate correctly

Link to all the inputs and outputs:

https://drive.google.com/open?id=1xOJXH3l0XrvD-Ht7zLqtr1TlrHopQL_c