This is a group (**maximum** 4 persons) assignment that will contribute 25% to your final grade.

Task A: YouTube Video Processing (14%)

One of your friends has just started working as a YouTube content creator for a week. At this point in time, she has recorded four videos from four different locations. However, as a content creator, she knows that she will need to capture and create more interesting videos. Since you are currently taking a course related to image processing, she has requested your help to develop a program that can be used to perform the following tasks automatically (or with minimal human intervention).

1) Detect whether a video is taken during daytime or nighttime (e.g., based on average brightness, histogram, etc.). If a video is found taken during nighttime, increase the brightness of the video before moving to the next step. You can decide the amount of brightness that should increase.
2) Blur all the faces (camera facing) that appear in a video.
3) Resize and overlay the video that she talks about her life as a YouTuber (talking.mp4) on the top left of each video. You can decide the location to overlay the video.
4) Add different watermarks (watermark1.png and watermark2.png) to the videos to protect them from being stolen.
5) Add the end screen video (endscreen.mp4) to the end of each video.

You can refer to the sample output (part_a_sample.avi) on eLearn. The sample output is produced using street.mp4, talking.mp4, endscreen.mp4, watermark1.png, and watermark2.png.

Hint/Guide:

The videos provided to you are recorded in 30 frames per second. In other words, there are 30 frames (or images) in one second. As shown below is how you could (i) read the frames from a video in a frame-by-frame basis, and (ii) save all the processed frames into a new video by using the built-in function from OpenCV.

```
vid = cv2.VideoCapture("street.mp4")                    #Read video.
out = cv2.VideoWriter('processed_video.avi',            #Set the file name of the new video.
                      cv2.VideoWriter_fourcc(*'MJPG'),  #Set the codec.
                      30.0,                             #Set the frame rate.
                      (1280,720))                       #Set the resolution (width, height).
total_no_frames = vid.get(cv2.CAP_PROP_FRAME_COUNT)     #Get total number of frames.
for frame_count in range(0, int(total_no_frames)):      #To loop through all the frames.
    success, frame = vid.read()                         #Read a single frame from the video.
    # Do something here.                                #Do something here.
    out.write(frame)                                    #Save processed frame into the new video.
```

Before blurring the faces, it is necessary to first detect and locate all the faces in a frame. As shown below is how you could (i) detect and locate faces from a frame by using a Haar feature-based cascade classifier, and (ii) draw a bounding box around them.

```
face_cascade = cv2.CascadeClassifier("face_detector.xml")   #Load pre-trained Haar cascade model.
faces = face_cascade.detectMultiScale(frame, 1.3, 5)        #Perform face detection.
for (x, y, w, h) in faces:                                  #To loop through all the detected faces.
    cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)  #Draw a bounding box.
```

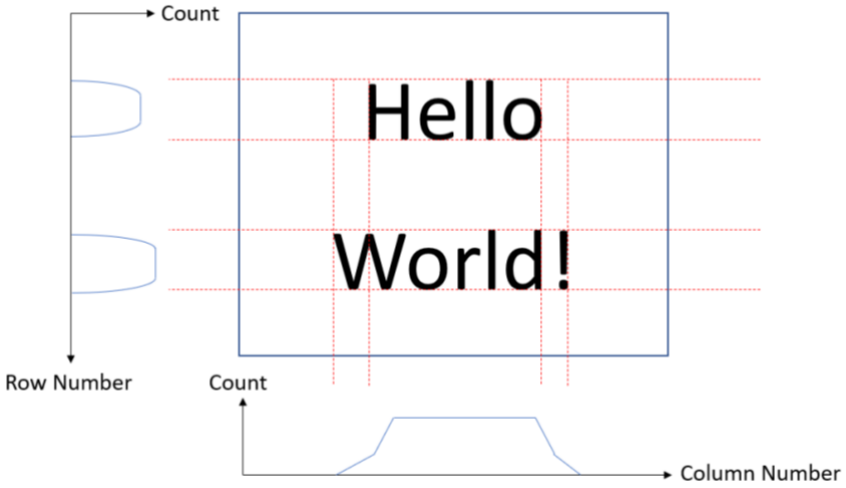
Task B: Paragraphs Extraction (11%)

One of your seniors is currently working on the literature review for his final year project. He has collected a set of scientific papers with different formatting (single-column, double-column and triple-column) from the Internet. His final year project topic is about applying the writing style of different authors. He has requested your help to (i) identify and extract all the paragraphs from the papers he collected, and (ii) sort them in the correct order (required for double-column and triple-column papers). Knowing that you are currently taking a course related to image processing, he has converted all the papers to images (001.png to 008.png) for you to apply the different image processing techniques you learned from the course.

You can refer to the sample outputs (or paragraphs) extracted from 008.png on eLearn.

Hint/Guide:

Histogram projection is a technique that can be used to identify the location of certain object (e.g. word). But the histogram here is slightly different from the histogram that we have covered. In this case, the histogram is used to record, for example, the number of black colour pixels found in each row or column of an image. Based on the two histograms shown below, we can identity the starting row and ending row, as well as the starting column and ending column of each word, by referring to row and column with number of black colour pixels more than 0.



Packages / Libraries

You are only allowed to use the following packages or libraries to complete the two tasks:

1) Python Standard (List)
2) OpenCV
3) NumPy
4) Matplotlib

Important:

1. All the code must be properly commented. Otherwise, marks will be deducted.
2. The due date of this assignment is 15th December 2023, 11:59PM.
3. You are required to submit a report to Turnitin (a link will be created on eLearn for this).
4. You are also required to submit another copy of your report, and your code, to eLearn.
5. Prepare the report in the following format using Microsoft Word with not more than 8 pages:
   - Single-Column
   - Single-Spacing
   - Arial
   - Font Size 12
6. You may include diagram(s) that could assist you in explaining your approach in the report.