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

Task A: YouTube Video Processing (12.5%)

One of your friends has just started to work as a YouTuber for a week. She has captured some random videos from different places. 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.

1) Blur all the faces that appear in the following videos automatically.
   a. street.mp4 (1280 x 720 @ 30 frames per second)
   b. exercise.mp4 (1920 x 1080 @ 30 frames per second)
   c. office.mp4 (1920 x 1080 @ 30 frames per second)
2) Resize and overlay the video that she talks about her life as a YouTuber (talking.mp4) on the top left corner of each video.
3) Add different watermarks (watermark1.png and watermark2.png) to the videos to protect them from being stolen.

Write a Python program to complete the task. You are not allowed to complete this task by using a video editing software. You can refer to the sample output/expected outcome (part_a_sample.avi) produced using street.mp4, talking.mp4, watermark1.png and watermark2.png on eLearn.

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 functions 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, total_no_frames):              #To loop through all the frames.
    success, frame = vid.read()                            #Read a single frame from the video.
    frame = cv2.add(frame,10)                              #Increase brightness by 10.
    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. Remember to first download the pre-trained Haar cascade model from eLearn.

```
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 (12.5%)

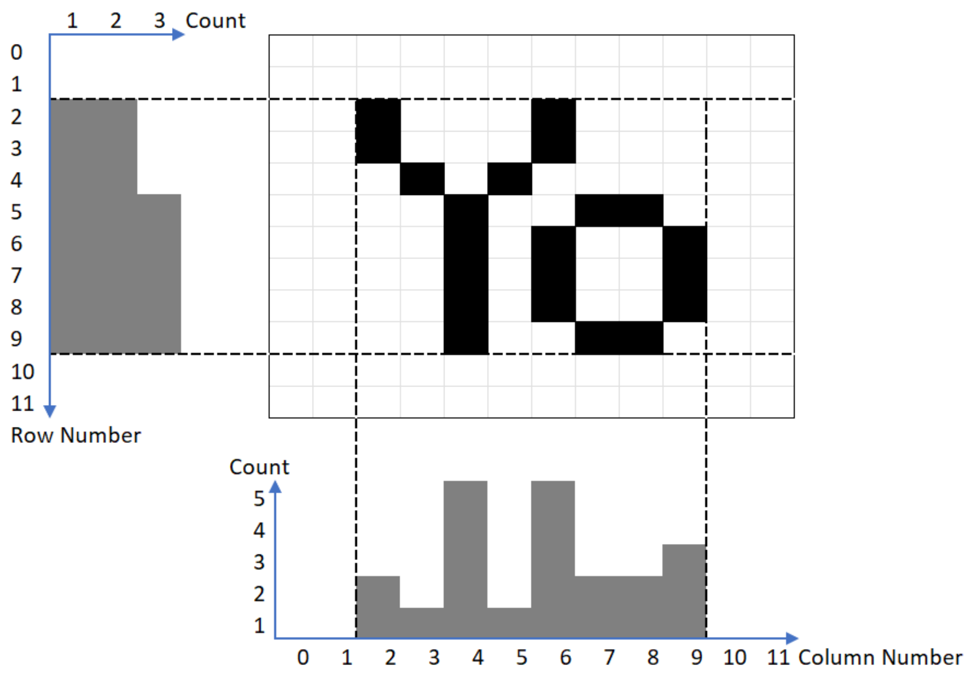
One of your seniors is currently working on the literature review for his final year project. He has collected a set of one-page scientific papers with different formatting (single-column, double-column and triple-column) from the Internet. His final year project is about analysing the writing style of different authors. He has requested your help to (i) identify and extract all the paragraphs from the papers he collected, (ii) sort them in the correct order (required for double-column and triple-column papers), and (iii) ignore tables in the papers (i.e. do not extract the tables).

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 different image processing techniques you learned for the course.

Write a Python program to complete the task. You can refer to the sample outputs/expected outcomes (part_b_sample_outputs) produced from 002.png on eLearn.

Hint/Guide:

Histogram projection is a technique that can be used to identify the location of certain object (e.g. word, paragraph, etc). 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 the word "Yo" by referring to row and column with number of black colour pixels more than 0.



Although the example demonstrated here is about identifying and locating the location of a single word, you can apply similar concept (with some extra algorithms) to identify, locate, and extract the paragraphs from those papers (or images) given to you.

Deliverables

1) Program codes written for the two tasks mentioned above.
2) A single-column report with less than 3000 words. The report should include the following sections:
    a. Introduction
    b. Methodology / Proposed Approach
    c. Results and Discussions
3) A demonstration video with not more than 5 minutes to explain and show the (i) execution of your program codes, and (ii) how to access and read the results produced by your program codes.

Packages / Libraries

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

1) Python Standard (Link)
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 21st November 2022, 11:59PM.
3. You are required to submit your report to Turnitin (a link will be created on eLearn for this).
4. You are also required to submit another copy of your report, together with your program codes and the demonstration video to eLearn.
5. Prepare the report in the following format using Microsoft Word:
    - Single-Spacing
    - Arial
    - Font Size 12
6. You may include diagram(s) that could assist you in explaining your approach in the report.