Date: 11th Oct, 2021 Sai Shashank GP

Technical Assignment

Domain: Computer science
Object Oriented programming

Problem Statement:
Using Python programming, pass an image acquired by webcam via queue from one process to another with the help of multiprocessing or multithreading.
Process 1: Image acquisition via webcam.
Process 2: Image reception and showing.
Process 3: Apply an OpenCV algorithm of my choice.

My Approach:
Since OpenCV is a total new area to me, I tried to understand about it as much as I could in this time. I followed a YouTube video by freecodecamp.org[1]. Once I got to know about how to implement the processes 1 & 2 in the form of code, I started coding them.
I started to test them locally and then ran the code in Google colab. But when I wanted to use **`cv2.imshow()`** in the colab notebook, it threw an error and it showed the alternate solution. It suggested to import **`from google.colab.patches import cv2_imshow`**. I used it but there was one more problem, Google colab cannot directly access the webcam on local machines. So, I searched the internet and found a stack overflow website[2] that it is already programmed and one should insert that code to access webcam. It also comes with a 'capture' button which captures the current frame when it's pressed and stores it as 'Photo.jpg' in the working directory. After all this hassle, I decided to stick to my local machine and upload code in GitHub after successful testing.

I first did not use any classes, functions or processes but just coded it raw and checked whether its working or not. I wanted to learn basics about multithreading and multiprocessing. So, I followed this YouTube video playlist for multithreading[3] and this YouTube video for multiprocessing[4]. After this, I started building a class for each process and defining some class methods. I decided to go with threading because I thought I was comfortable with it. In the initial try, I was neither able to show webcam video nor able to take screenshot. It threw an error. After a bit research in stack overflow, according to this answer, OpenCV doesn't support threading and only multiprocessing is to be used[5]. So, I started to use multiprocessing.

---

[1] https://www.youtube.com/watch?v=oXlwWbU8l2o&t=1214s

[2] https://stackoverflow.com/questions/54389727/opening-web-camera-in-google-colab

[3] https://www.youtube.com/watch?v=GFfMuyNUnDs&list=PLzMcBGfZo4-mL_4mo5LbOIdPV8jQ-n4ib&t=0s

[4] https://www.youtube.com/watch?v=fKl2JW_qrso&t=1756s

[5] https://stackoverflow.com/questions/66245941/how-to-use-multithreading-with-and-cv2-videocapture

According to the multiprocessing video I mentioned before, there is a class in the module **`concurrent.futures`** called **`ProcessPoolExecutor()`** which is used widely and is somewhat new compared to **`multiprocessing`** module. So, I used it and achieved the processes 1 & 2 successfully. It ran on local machine. It won't be possible to make this code run on Google colab. This GeeksforGeeks page has some basic info about the **`ProcessPoolExecutor()`** class[6] .

Useful links:

GitHub repo for code-

Contact details:

Sai Shashank GP
2nd year B.Tech in EE dept. at IIT Madras
Mobile: +91 99513 67352
Email: **ee20b040@smail.iitm.ac.in**
Alt.email: **saishashankgp2003@gmail.com**
LinkedIn: **https://www.linkedin.com/in/sai-shashank-gp-466617217/**

[6] https://www.geeksforgeeks.org/processpoolexecutor-class-in-python/