

## Task01

### Question 2(Bounding Box):

The goal of this project was to gain a deeper understanding of manual annotations and their use in creating datasets. To accomplish this, a dataset of images featuring the faces of all group members was created. Subsequently, the OpenCV library was utilized to write code for retrieving the coordinates of the two opposing corners of the bounding box surrounding each face. These coordinates were then saved in a pickle file for future reference. Finally, a Gaussian Blur function was implemented to apply blur to specific parts of the image.

Here is the result of the code on a random image taken from the internet.



Original Image



Anonymised Image

More explanation of the code is also provided in the code in the form of comments. Our main learning from this question are as follows:

1. Using `setMouseCallback` function of `opencv`.
2. `Pickle` library of `python`.
3. `Argparse` library of `python`.

#### Documentation of code:

Images can be annotated using the following command:

```
python bounding-box.py --data <path-to-folder-containing-images> \  
--annotation <path-to-save-annotation> --type 1
```

After running the command in terminal an image will be displayed. You can annotate it using a mouse drag.(P.S. Don't click to select basics, drag from one corner to other pressing left mouse button). Press 'q' to proceed.

### Question 3(Edge and Contour Detection):

In this task we implemented 2 different functionalities, edge detection and contour detection, for the edge detection we used `canny` - a opencv library, to detect edges, we learned that its not easy to get the outer edges(i.e., edges on the outer border of butterfly in this case) properly, because we saw that using library it detects so many smaller edges, which is away from our objective, during the 'edge detection method using gaussian filter' topic, sir told that its not easy to get the outer edges. Hence we experienced it.

In the other task we basically got familiar with `findContours` - a opencv library, and created a program to detect max contour of an binary image. As asked in problem statement, we used our group member image and then binarized it manually using image editing softwares and performed the same task on that image and we got contours of our image. to find the contour of the tallest person, we select the contour having maximum distance between top-most point and bottom-most point. and we can find contours of all the objects of the in images using edge detection library only if the images are binary.

### References:

- a. [https://docs.opencv.org/4.x/d7/d4d/tutorial\\_py\\_thresholding.html](https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html)
- b. <https://learnopencv.com/read-write-and-display-a-video-using-opencv-cpp-python/>
- c. <https://www.tutorialspoint.com/opencv-python-how-to-draw-a-rectangle-using-mouse-events>
- d. <https://stackoverflow.com/questions/25464295/dump-a-list-in-a-pickle-file-and-retrieve-it-back-later>
- e. <https://docs.python.org/3/library/argparse.html>
- f. <https://stackoverflow.com/questions/18351951/check-if-string-ends-with-one-of-the-strings-from-a-list>
- g. chatgpt