The name of this task is ‘Who is the intruder?’, as the name suggests we have to find whether the intruder (i.e. student with lowest roll number) is present in the dataset or not.

- Step I – Add images (selfie) of all the group members in the dataset. For better matching we have removed the background. Masked picture of lowest group member should be added in the folder ‘capturedImages’.
- Step II – This step involves calibrating **‘reference retrieval score’**, this is score at which we can positively say that intruder is present in the dataset. There is no compulsion as to what metric to be chosen for setting reference retrieval score.
- First we detected set SIFT to retain 1500 features for the masked image and then iterated through all the images in the dataset.
- We then use a two different strategies to increase the matches one being BruteForce matcher with L1 norm and with set cross checking, and other being a knn based matching that uses ratio test (We think from Lowe’s paper). This significantly increases the number of matches.
- Next we use Homography to separate out outliers. We used RANSAC algorithm setting the reprojection error threshold to 10.
- This gives us new set of matches which are free of most outliers. On top of this, we have added another filtration strategy in which we perform the homography for the points (Hp1 and p2 where H is the homography matrix, p1 is point from img1 masked image and p2 is point from an image from database) we have for matching. If the L2-Norm of between this transformed is less than a threshold (set to 3.5 pixels) then we call this a “good match”. The metric we use to set the retrieval threshold is the ratio of the number of “good matches” to the number of total matches after homography.
- (The retrieval score/threshold we decided to stick to is 0.65).
- Step III – Once the reference retrieval score is set at **‘0.65’**, next and main task is to find out how much masking is allowed to achieve calculated score. For this part, group member middle’s face image has to be matched with the same image but with mask. To draw mask we used opencv’s circle function, by varying\increasing the radius of circle we get more face exposed in the image. Finally we have kept increasing radius by factor of ‘15’ and did the step explained in ‘Step II’ to achieve desired score. **Result** is as followed:

  We achieved desired score at Diameter value equal to 100.

  Note that the unit of radius in opencv’s function is number of pixel along the radius. So when we say diameter=100 , we meant 50 pixels along the radius.

  As an aside, we also provide setting as mentioned in the Task pdf, it is assumed that we have provided Middle.jpeg (to match the masked image with) in data/replicate/capturedImages directory. We give an automated option as described above, more details about this is present as comments in the code.