### **CS517 PA2-Image Morphing**

**Name** - Deepak Kumar

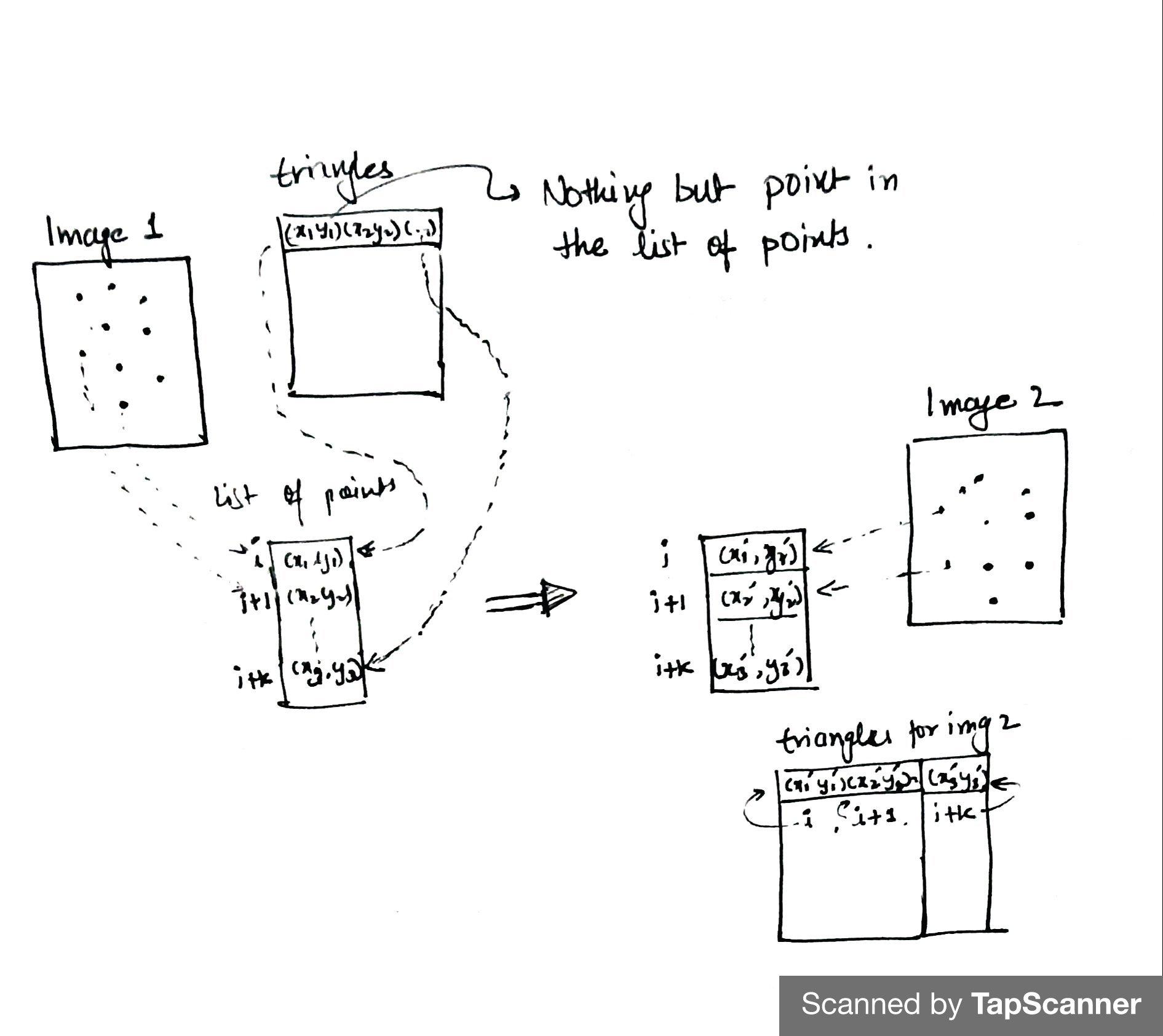
**Entry no.** - 2022CSM1002

**Required packages:**

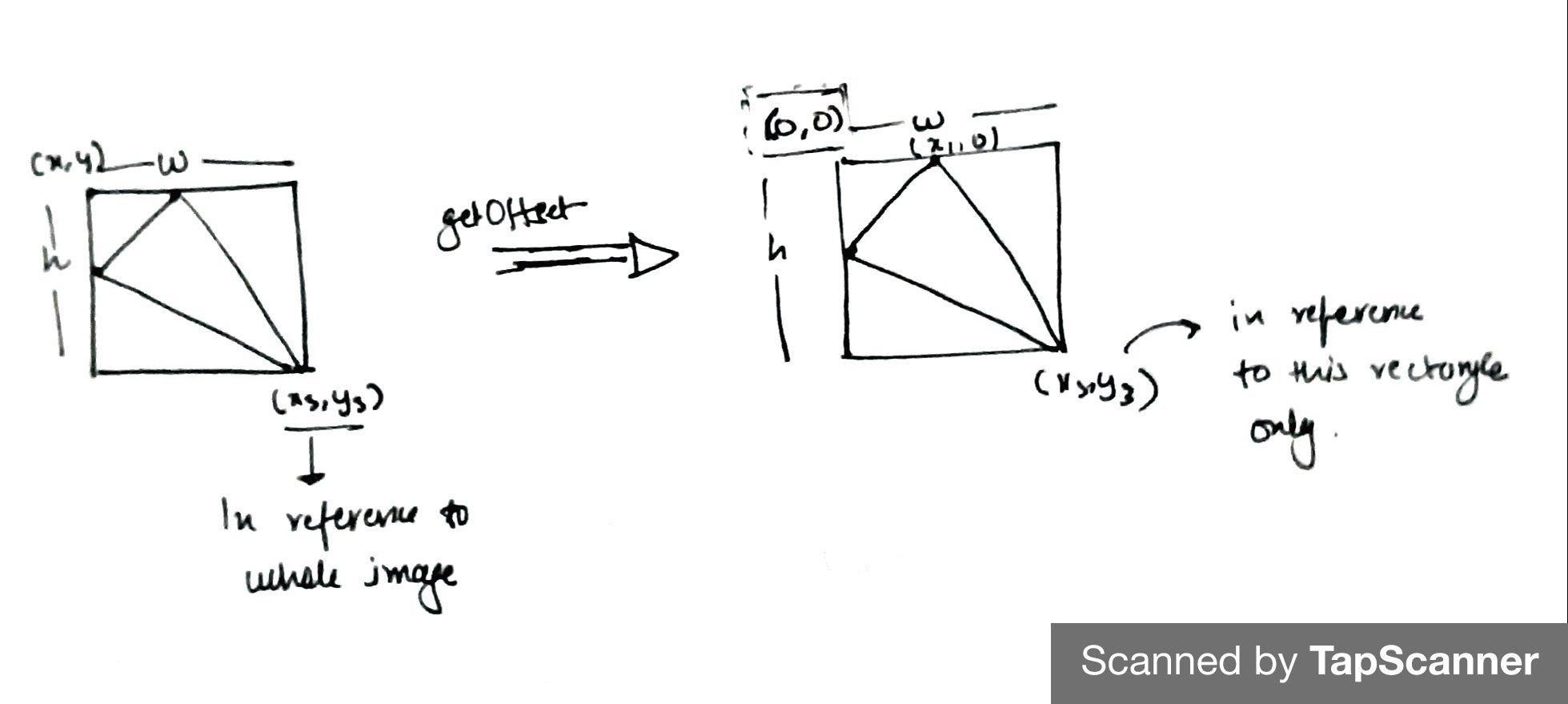
* import imageio
* from imutils import face\_utils
* import dlib
* import cv2
* import matplotlib.pyplot as plt
* import numpy as np
* import sys
* from IPython.display import Image

**Methodology:**

* Feature detection using dlib or manual points for both the corresponding images, i th index in both lists of points corresponds to the same feature.
* We have points for both source image and destination image using dlib, to find points in morphed image the points will vary according to alpha value, so for **x\_ = (1 - alpha) \* x1 + alpha \* x2, y\_ = (1 - alpha) \* y1 + alpha \* y2**.
* Next is triangulation, for triangulation subDiv2d is used, only triangulation of source image (image 1) is done using subDiv2d for the triangulation of morphed image and destination image (image 2) we require corresponding triangles according to the i th index so what i done is, we have list of triangles for the source image. Let’s say we have triangle 1 with coordinates [[x1 y1] [x2 y2] [x3 y3]] to find triangle 1 for our destination image, I will find index of [x1 y1] [x2 y2] [x3 y3] in the list of points for source image then using those indices i will pick my points from list of points i have for destination image which will give me the corresponding triangle in destination image, similarly for morph image.

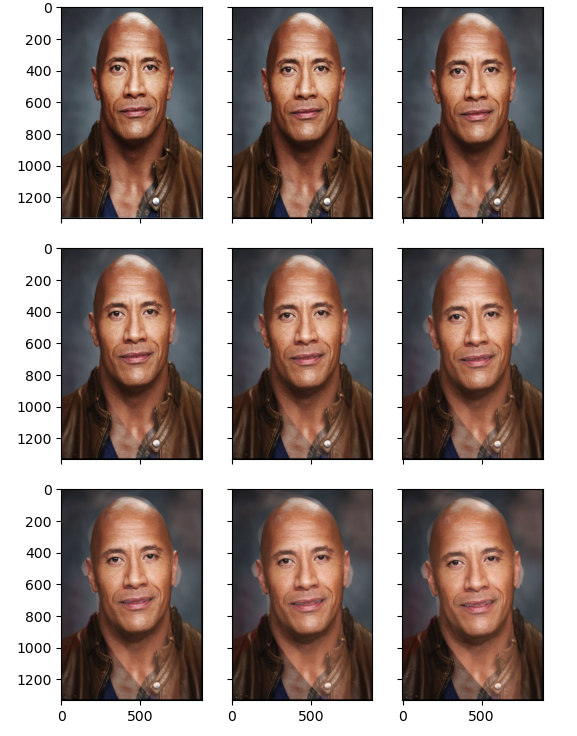


* Now i am doing image warping by applying affine transformation but warp affine will be applied by taking small rectangular regions bounding the triangle so affine transformation which will be in reference of that rectangle that’s why i am taking offset of triangles without this resulting coordinates may go out bounds and taking a mask to later extract that triangle form the rectangle so that background pixels won’t get affected using this we will get warpImage1 and warpImage2.



* Now I have warpImage1 which I got using affine transformation between triangles of src image and morphed image and warpImage2 using affine between destination and morphed.
* Now I am doing alpha blending to adjust the intensities of the pixels in the morphed image, **morphTr = (1.0 - alpha) \* warpedWithDwayne + alpha \* warpedWithVin**

**Result:**

****