Image Processing: Digital Assignment 2 & 3

Shreyansh Saha, 17BCE1148

Abstract

This assignment tries various methods to calculate similarity between (1) Images of individual at different age (2) Images of Individual and their family members. It demonstrates methods with fixed scale for similarity measurement as well as methods with no defined scale and give comparison between all the methods.

It uses 68 Facial Points Annotation [1][2], Manhattan Distance, Earth Mover's Distance [7], SSIM and SIFT Similarity measure [10].

1. INTRODUCTION

In my survey, I was not able to find an algorithm to detect similarity between faces except SIFT (Scale-Invariant Feature Transformation) and Facial recognition algorithm using Neural Network. Since I only had a limited number of images I could not train a neural network and used the SIFT Algorithm along with some works of my own which I made by improving upon '68 facial landmarks' to detect facial features. I have also used Earth Mover's distance to compare histograms of two images.

I detected facial features using the dlib library in python and compared the distance between the facial landmarks of two images. I have scaled and normalized the images as needed to rectify the errors due to difference in resolution.

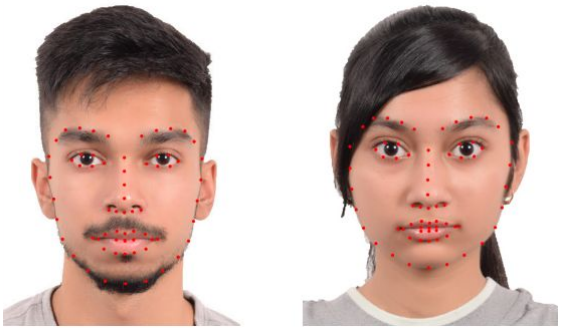

Figure 1. Using 68 landmarks to map facial features.

2. MAPPING FACIAL FEATURES

Using Dlib in python first we need to import the predictor. Dlib also has a face detector which we can use to isolate the region of interest and then map the landmarks.

Code Snipped to get landmarks in the ROI:

```
def getAndMapFeatures(gray, image, rects):
  points = []
  for (i, rect) in enumerate(rects):
    shape = predictor(gray, rect)
    shape = face_utils.shape_to_np(shape)
    points = shape
    for (x, y) in shape:
      cv.circle(image, (x, y), 1, (0, 0, 255), 2)
    return image, points
```

Using this we get the following landmarks.

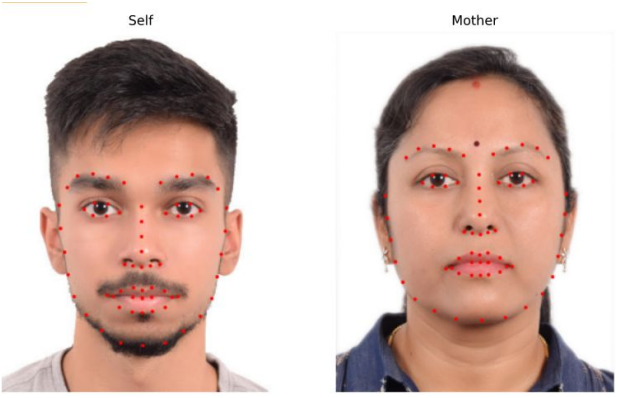

Figure 2. Landmarks between self and mother

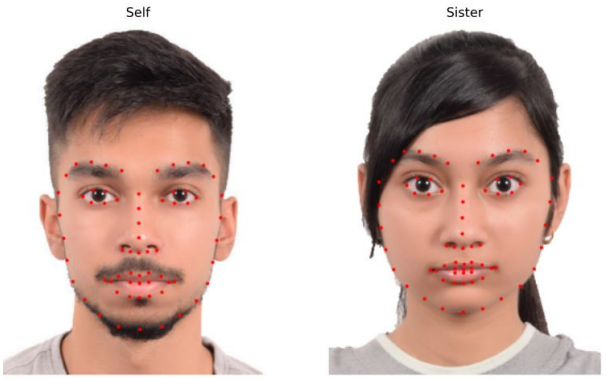

Figure 3. Landmarks between self and sister

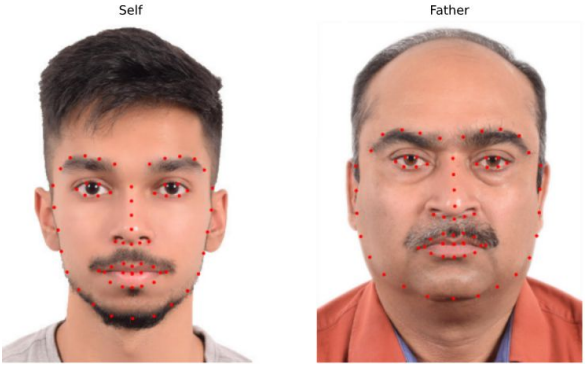

Figure 4. Landmarks between self and father

3. SCALING LANDMARK POINTS

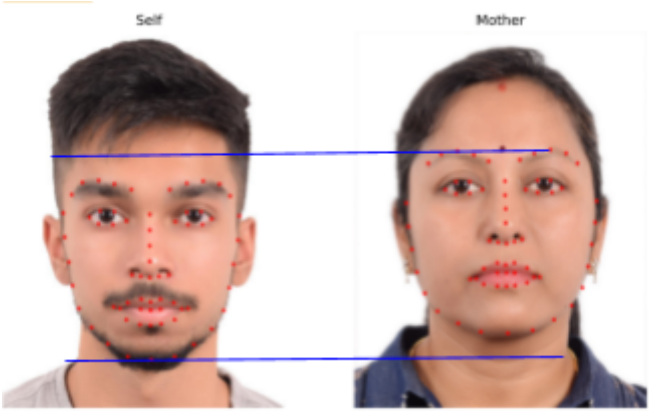

Figure 5. Difference in range of points

We are concerned with distance between facial features and not the starting and end location of those features. For this reason we will scale the features by subtracting all x and y points with their minimum value. This will ensure all points start at (0, 0) and distance between landmarks is preserved. Since the resolution of all these images is the same, this scaling is enough to get proper results.

Code Snipped to scale the points:

```
def scalePoints(points: NumpyArray)->NumpyArray:
    # Lowest x and y coordinates
    xl, yl = np.min(points[:, 0]),
             np.min(points[:, 1])
    m, n = points.shape
    for i in range(points.shape[0]):
        points[i][0]-=xl
        points[i][1]-=yl
    return points
```

We scale the points and get the following landmarks.

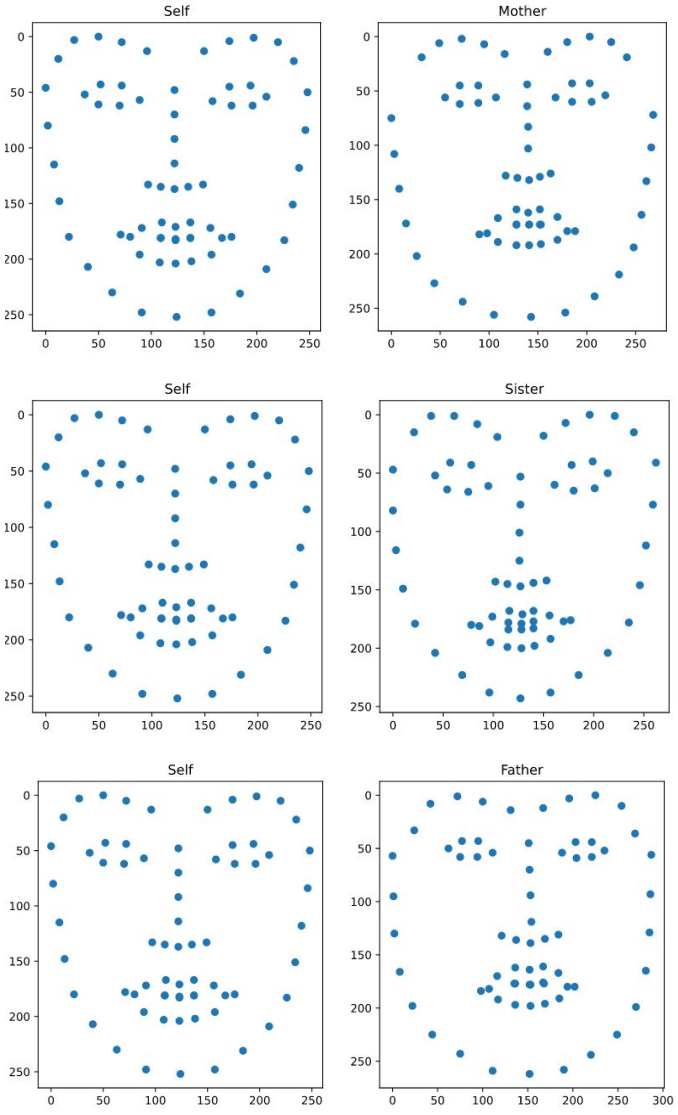

Figure 6. Landmarks points between self and family members

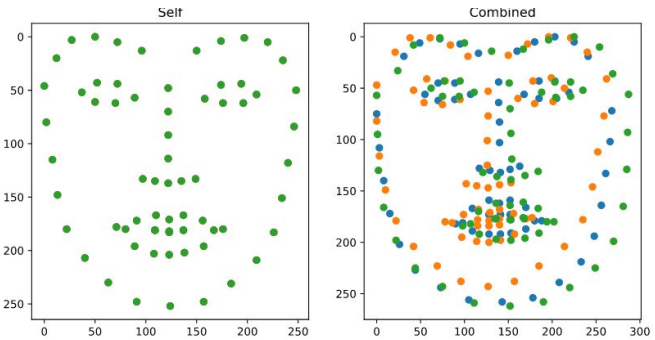

Figure 7. Landmarks between self and combined family image

4. MANHATTAN DISTANCE

We now calculate the Manhattan distance between these landmark points to rank the images based on similarity to 'self' image.

Manhattan Distance:

$$\frac{1}{m*n}\sum_{i}^{m}\sum_{j}^{2}|x_{ij} - y_{ij}|$$

m = Number of points
x = array of points (a, b)
y = array of points (a, b)

Result

Compare	Self
Self	0.0
Mother	11.382353
Sister	4.610294
Father	16.102941

Lower value denotes higher similarity. This method cannot give us a similarity in percentage between 0-100. It can only compute the distance between landmark points between the two images.

We can use this to compare multiple images with a single image and rank images based on similarity.

5. EARTH MOVER'S DISTANCE

$$\mathrm{EMD}(P,Q) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij}f_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}},$$

d = distance
f = flow

Earth Mover's distance can tell us how far one histogram is from another histogram. It calculates the distance needed to make one histogram equivalent to another. Using SIFT algorithm we see that we still get the same

We calculate the histogram (gray) of all images and calculate the Earth Mover's distance.

Result

Compare	Self
Self	0.0
Mother	0.000810
Sister	0.000378
Father	0.001752

Similar to Manhattan distance, EM distance cannot tell us how much an image is similar to another image. It can be used to rank images based on the histogram. As we can see, the ordering of similarity is the same as with Manhattan distance.

6. SIFT SIMILARITY

Scale-Invariant Feature Transform (SIFT) can work on multichannel images with different resolutions and scaling.

SIFT works by detecting local features in images. We then compare these features between two images to compute similarity.

Result

Compare	Self
Self	1.000000
Mother	0.983607
Sister	0.984962
Father	0.959459

As we can see SIFT works differently than the algorithms presented before. SIFT gives output in range [0, 1] where values closer to 1 denote higher similarity and values towards 0 denote less similarity.

ordering or similarity in images i.e.

Self > Sister > Mother > Father

7. SSIM

Structural Similarity Index (SSIM) is used to measure image degradation. However, it can be used to measure image similarity as well [11].

$$SSIM(x,\ y)\ =\ \frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)}$$

SSIM can also with multichannel images and gives output in range [0, 1] where 0 means low similarity and 1 means high similarity. It doesn't work with images of different resolutions.

Result

Compare	Self
Self	1.000000
Mother	0.538413
Sister	0.562699
Father	0.460827

Again we see that the order of similarity remains the same.

8. COMPARING SELF IN DIFFERENT AGES

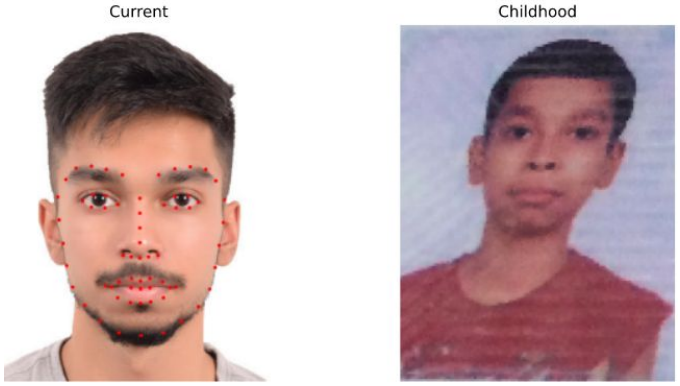

Figure 8. Images of current age and childhood age Code Snipped to normalize the points.

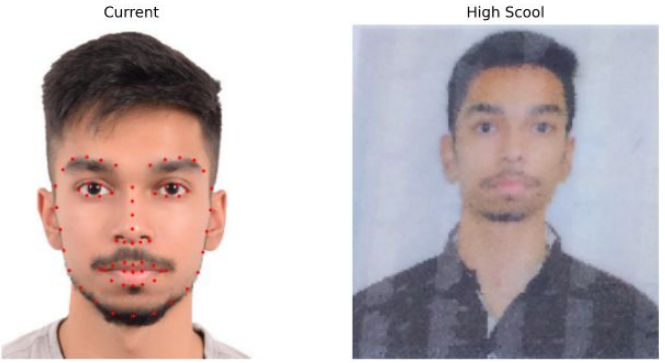

Figure 9. Images of current age and childhood age

As we can see the resolution as well the orientation is different for these images. We cannot use the scale operation we used above as the distance between features of both images is different.

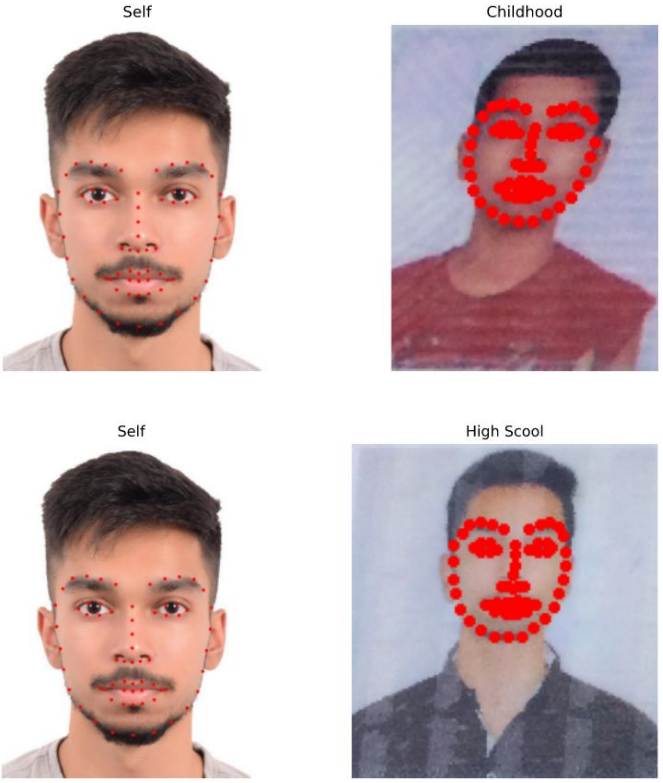

Figure 10. The facial landmarks on right image are skewed and have smaller distance between them as compared to 'Self' Image

To solve this we need to normalize the image to have a value between [0, 1].

We normalize the points by their min and max distance value.

```
def normalizePoints(points):
    x = (points[:, 0] - min(points[:, 0]))
        /(max(points[:, 0])- min(points[:, 0]))
    y = (points[:, 1] - min(points[:, 1]))
        /(max(points[:, 1])- min(points[:, 1]))
    x = np.array(x)
    y = np.array(y)
    return np.vstack((x, y)).T
```

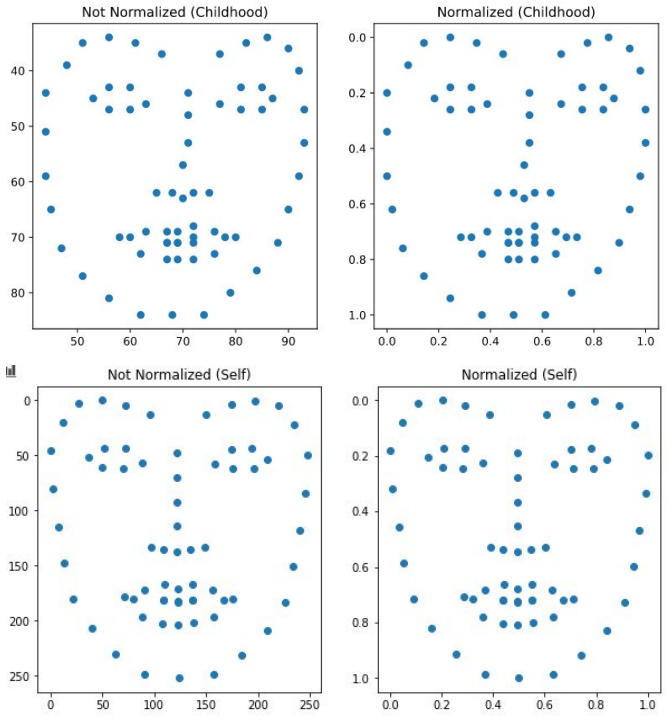

Figure 11. We can see that the distance between the landmarks is preserved but the values are in range [0,1] for both images. We can also see that normalizing fixed some of the skewness in the 'Childhood' image.

Results

Metric	Childhood	High School
Manhattan	0.035612	0.023204
EM Distance	0.003296	0.003396
SIFT Similarity	0.72	0.9

9. FUTURE WORKS

1. Combining SIFT and 68 Facial Landmarks to detect better features.

2. Comparing Color values of the landmarks instead of just intensity values

3. Improving similarity metrics when the image is skewed or rotated at an angle.

10. REFERENCES

[1] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, 2014, pp. 1867-1874, doi: 10.1109/CVPR.2014.241.

[2] iBug facial points annotation, Imperial College London

[3] Shungang Hua, Guopeng Chen, Honglei Wei and Qiuxin Jiang "Similarity measure for image resizing using SIFT feature" December 2012 EURASIP Journal on Image and Video Processing 2012(1);DOI: 10.1186/1687-5281-2012-6

[4] Helen Dataset, University of Illinois

[5] Detecting facial landmarks using dlib

[6] Dlib Documentation

[7] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas "The Earth Mover's Distance as a Metric for Image Retrieval", Computer Science Department, Stanford University
Stanford, CA 94305

[8] "Earth mover's distance" A semantic measure for document similarity in semantic search

[9] Ebrahim Karami, Siva Prasad, and Mohamed Shehata "Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images"; Faculty of Engineering and Applied Sciences, Memorial University, Canada

[10] F. Guo, J. Yang, Y. Chen and B. Yao, "Research on image detection and matching based on SIFT features," 2018 3rd International Conference on Control and Robotics Engineering (ICCRE), Nagoya, 2018, pp. 130-134, doi: 10.1109/ICCRE.2018.8376448.

[11] The SSIM Index for Image Quality Assessment, New York University

[12] Image Classification using SSIM

Code and Research Papers available on github:
https://github.com/shreyanshsaha/IP_DA23

(Code is available in the form of jupyter notebook for easier readability)

Note: Repository is IP_DA23