COMPUTER VISION ASSIGNMENT

DATE DUE: 12/05/2022

CANDIDATE NUMBER: 219060

WORD COUNT:

Table of Contents

Abstract 0
Method 1
Variations in points 1
Average Points 2
Moving the Average Points 3
Results and Analysis 4
Changing Lip Colour 5
Method 5
Results and Analysis 6
References 8
Appendices 8

Introduction

This report details the processes used to create a face alignment tool with a modal average of 8 pixels distance between computed and labelled points. On top of this, a lip colour changer is also outlined

Method

Variations in points

I tried to do this assignment without the use of a neural network / machine learning of any form so that I could compare how well logical reasoning and searching would compare to machine learning techniques.

The brief states that "the eye corners [are] in the same position." [1] This was tested this to see if the location of the eyes could be trusted as fixed points that could then be used in calculations.

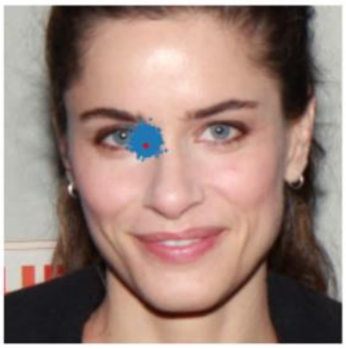

Figure 1

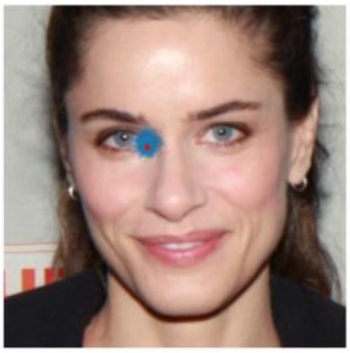

Figure 2

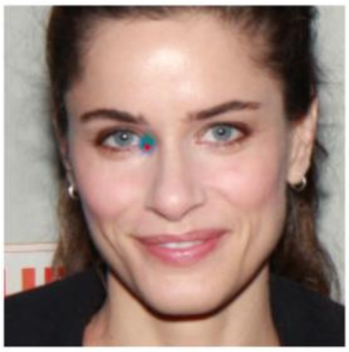

Figure 3

Figures 1-3 are scatter graphs of the marked location of the corner of the person's eye (point 19) with varying alpha values (1, 0.1 and respectively) over one of the images from the training dataset provided. The different alpha values in figures 2 and 3 are used to see the distribution and density of the points with the red dot being the location of point 19 that corresponds to the background image. If just the red dot is seen, all the points for the corners of the eyes are in the same position, however, as it can be seen from these figures, the locations of the eye corners were not in "the same position". The spread in figure 1 shows just how varied these points are.

Average Points

An average location for each labelled point was computed. The average and labelled locations for all points have then printed on the image (figure 4).

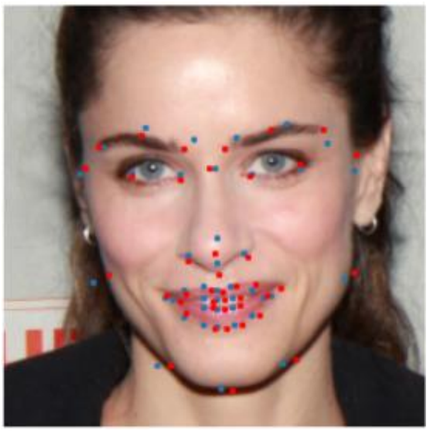

Figure 4

It can be seen that several of the average points are not in the same location as the labelled points. In order to overcome this, an edge map of the image was created using Canny on a greyscale image. The average points have then been moved to the closest edge. This is done by getting an array of distances to all the edges for each point. The edge location with the shortest distance to the average point is selected unless the closest edge is greater than the maximum distance the points are allowed to move.

It can be seen (figure 5) that, in many locations, the adjusted average points are closer to the labelled points than the average points

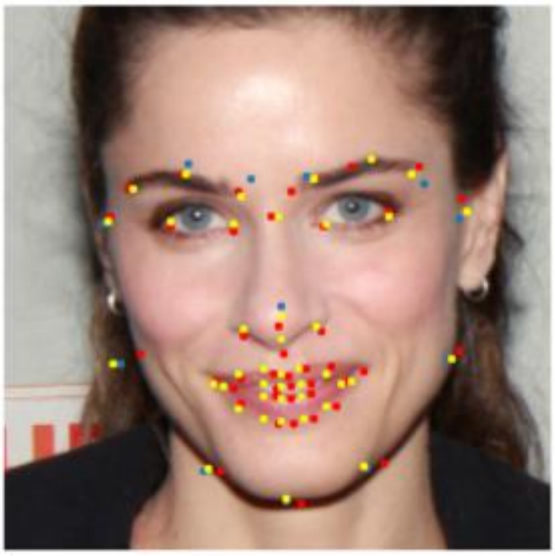

Figure 5

This figure displays the set of average points (blue), the set of labelled points (red) and the set of adjusted average points as described above (yellow).

Moving the Average Points

In some circumstances, moving one (or more) of the points too far will distort the location(s) of the point(s) making them less accurate than if they were not moved. One case where this may happen is: if the jawline is not obvious due to the background of the image being of a similar colour to the person's skin tone. With each point being allowed to move an unlimited distance from the average location of that point, this may lead to situations like lips to being described as the chin. The two main points that should not be on an edge are the tip of the nose and between the eyes.

A function was made that would measure the average distance between the labelled location of a point and the adjusted average location. The maximum distance that each set of points can move is limited to varying degrees in order to find the optimum upper bound for movement. In each set of points, the distance that the points which describe the jaw are allowed to move is different to those that describe the rest of the features, this is because I found most variation in the positions of the jawlines in the images.

The optimum upper bounds of movement for the points 4950 units for the jawline and 1490 units for the feature points. Some yellow points (figure 6) are closer to the labelled (red) points than some of the purple points. This is because the maximum distances were calculated on all the training images.

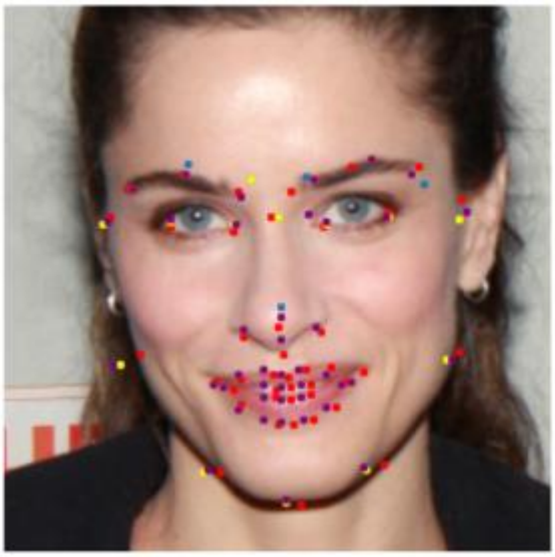

Figure 6

Figure 6 displays the set of labelled points (red), the set of average points (blue), the set adjusted points (yellow) and the adjusted points using the computed movement bounds (purple).

The only pre-processing that was done to the images is turning then greyscale before the edge detector was applied. The images were not smoothed as this would interfere with the edge detector, creating a less reliable edge map. The images sizes and resolutions did not need to be reduced as they were not passed through a neural network.

Results and Analysis

As seen (figure 7) the modal average of average distance between the adjusted points and the labelled points is 8 pixels. The mean average is 11.526.

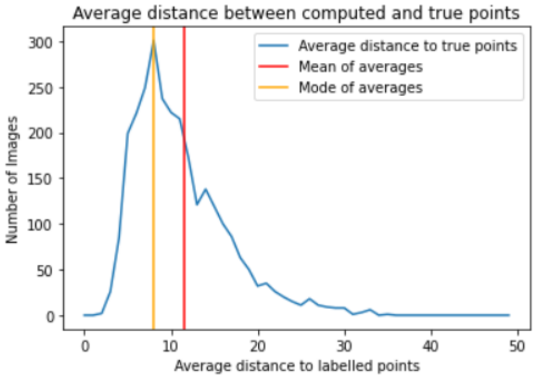

Figure 7

As the method used relies on the average location of points, there is an inherent bias where points can be more accurately marked if the person is looking directly at the camera. The scale of the face is also important when it comes to accuracy. The final attribute that effects the accuracy most is the amount of edges present. As the average points are relocated to the nearest edge in range and hair and wrinkles can cause a high-density edge map to be generated, if the computed average location of a point is wrong, there is a high likelihood that the point will either not move or will not move very far.

This sort of face detection would work quite well in controlled circumstances like with passport photos. As in a passport photo, the persons face needs to be head on, of a certain size and with hair tied back.

Changing Lip Colour

Method

One method considered for clanging lip colour was to get a rough average colour of the lips using pairs of points, e.g.: point 40 and point 31. A point would be generated between these points. The RGB values of all these new points would then be averaged to find the average lip colour. Values would be added and subtracted to the averages to create an upper and lower bound to the colours of the lips.

A mask would then be made based on this colour.

The difference between the average colour and the desired colour would then be computed to create a swatch to apply to the image using the mask so that the colour would only apply to pixels that the mask denoted as the lips.

I found that the accuracy of this method changed drastically with the person's skin tone. The areas that the mask detected using the computed average lip colour largely encompasses areas at the edges of the face (figure 8).

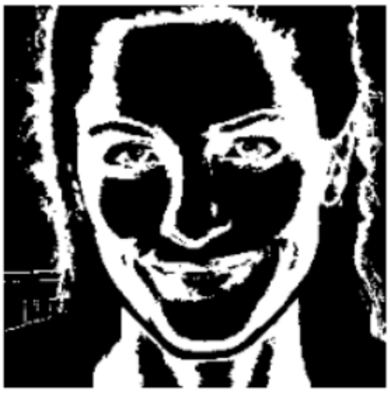

Figure 8

Mask of face using colour range of 50 from the average RGB value

When reducing the range of colours, the mask detects less area at the edge of the face but this also reduces the detected area of the lips.

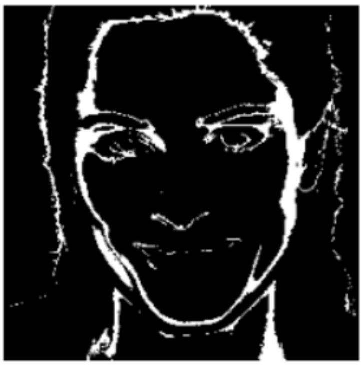

Figure 9

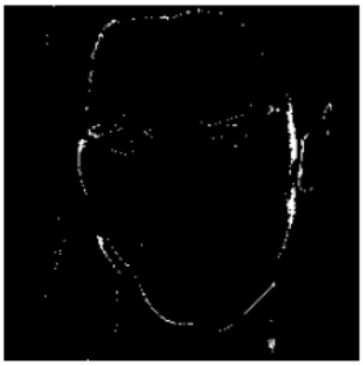

Figure 10

Figure 9 is a mask detecting pixels with a colour range of 25 either side from the average RGB value. Figure 10 uses a range of 10.

Due to the inaccuracy of this method, a different method was used. A blank mask was created on which a shape was drawn connecting all the points that constitute the lips. This chape was filled in to create a mask of the lips.

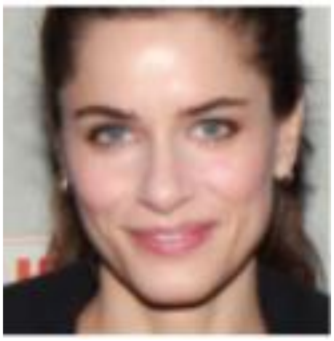

Figure 11

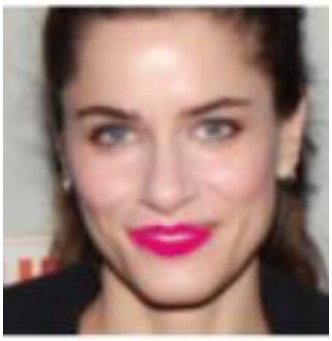

Figure 12

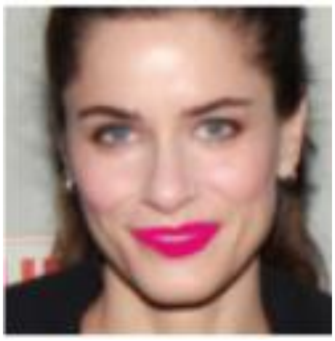

Figure 13

Figure 15 is the image the new lip colour is being applied to, figue 16 uses the mask based on the points generated and figure 17 uses the labelled points provided.

Results and Analysis

This method performs adequately but is unreliable due to its dependence on the accuracy of the labelling of the required points. On top of this, when the desired lip colour is black, this method outputs an entirely black image (figure 14). This issue was solved by inverting the mask when the desired lip colour is black.

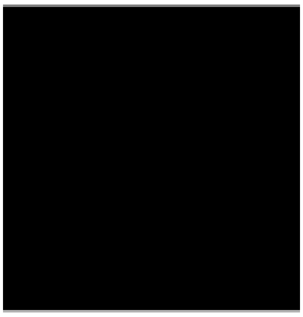

Figure 14

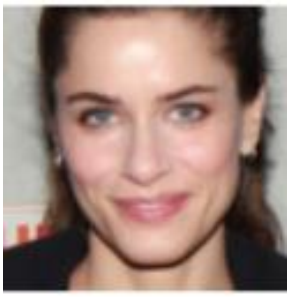

Figure 15

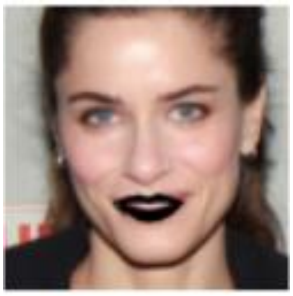

Figure 16

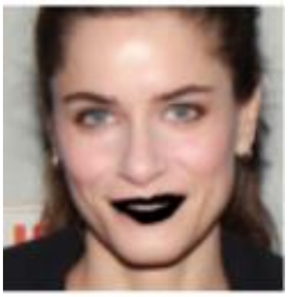

Figure 17

Figure 15 is the image the new lip colour is being applied to, figue 16 uses the mask based on the points generated and figure 17 uses the labelled points provided.

Only minor amounts on the texture lips is preserved using the method due to the colour of the pixel being set to an average of the sum of the new and original colours

One way to preserve the texture of the lips would be to apply SIFT on the image using the mask. The average of these colours could then be adjusted to give the desired outcome.

References

[1] I Simpson. Coursework Assignment. University of Sussex – Spring 2022 (pp. 3). Available at: https://canvas.sussex.ac.uk/courses/17639/assignments/84119

Appendices

Python Pillow - Blur an Image. (2021, April 6). GeeksforGeeks. [ONLINE] Available at: https://www.geeksforgeeks.org/python-pillow-blur-an-image/ [Accessed: 06 May 2022]

Python PIL | ImageDraw.Draw.polygon() Method. (2019, August 21). GeeksforGeeks. [ONLINE] Available at: https://www.geeksforgeeks.org/python-pil-imagedraw-draw-polygon-method/ [Accessed: 06 May 2022]

skimage.draw.polygon Example. (n.d.). Programtalk.com. [ONLINE] Available at: https://programtalk.com/python-examples/skimage.draw.polygon/ [Accessed: 06 May 2022]

Python OpenCV | cv2.line() method. (2019, August 7). GeeksforGeeks. [ONLINE] Available at: https://www.geeksforgeeks.org/python-opencv-cv2-line-method/ [Accessed: 06 May 2022]

Python PIL Tutorial. (n.d.). HolyPython.com. [ONLINE] Available at: https://holypython.com/python-pil-tutorial/ [Accessed: 06 May 2022]

Draw.FillPolygon - OpenTuring Documentation. (n.d.). Documentation.help. [ONLINE] Available at:

https://documentation.help/OpenTuring/draw_fillpolygon.html [Accessed: 07 May 2022]

Coelho, L. P. (n.d.). *mahotas: Mahotas: Computer Vision Library*. PyPI. [ONLINE] Available at: https://pypi.org/project/mahotas/ [Accessed: 07 May 2022]

Plot types — Matplotlib 3.5.2 documentation. (n.d.). Matplotlib.org. [ONLINE] Available at: https://matplotlib.org/stable/plot_types/index.html [Accessed: 07 May 2022]

Python PIL | getpixel() Method. (2019, July 15). GeeksforGeeks. [ONLINE] Available at: https://www.geeksforgeeks.org/python-pil-getpixel-method/ [Accessed: 07 May 2022]

Python Numpy – Array Average – average() - Python Examples. (n.d.). Pythonexamples.org. [ONLINE] Available at: https://pythonexamples.org/python-numpy-average/ [Accessed: 06 May 2022]

pytutorial | Python: How to get the type of a variable. (2022, March 5). Pytutorial.com. [ONLINE] Available at: https://pytutorial.com/python-how-to-get-the-type-of-a-variable [Accessed: 07 May 2022] *numpy.matmul — NumPy v1.22 Manual*. (n.d.). Numpy.org. [ONLINE] Available at: https://numpy.org/doc/stable/reference/generated/numpy.matmul.html [Accessed: 06 May 2022]

numpy.full — NumPy v1.19 Manual. (n.d.). Numpy.org. [ONLINE] Available at: https://numpy.org/doc/stable/reference/generated/numpy.full.html [Accessed: 06 May 2022]

numpy.round_ — NumPy v1.22 Manual. (n.d.). Numpy.org. [ONLINE] Available at: https://numpy.org/doc/stable/reference/generated/numpy.round_.html [Accessed: 07 May 2022]

numpy.round_() in Python. (2018, April 9). GeeksforGeeks. [ONLINE] Available at: https://www.geeksforgeeks.org/numpy-round_-python/ [Accessed: 07 May 2022]