Coursework: Face Alignment, Segmentation and Graphical Effect

215758
G6032: Computer Vision

May 13, 2021

Abstract

The face alignment problem is about locating facial landmarks where you need to locate coordinates to describe the shape and pose of someone's face. This problem is difficult to solve because of the amount of variation of face. Our method is using a convolutional neural network (CNN) model paired with preprocessing utilising histogram of oriented gradients to make a feature vector for the CNN. The result was 86% on the test data split, this is good however not good enough for something which would be in production in our view. We also achieved facial segmentation however it is dependent on the facial landmarks. Additionally we made a graphical effect from the landmarks which gave good results however was also dependent on the facial landmarks. This was using a very basic model, this could be a factor to further improve CNN based methods to solve the face alignment problem.

1 Introduction

Face alignment, also known as locating facial landmarks in images, is where a set of coordinates describe the shape and pose of the face[2]. This problem can be difficult due to the varying positions to predict.

We will be using a Convolutional Neural Network (CNN) model with Histogram of Oriented Gradients (HOG) as a preprocessing step. A CNN is a deep learning neural network used for processing structured data, an example of such are images[6]. It can be used for classifying letters from handwriting as an example. HOG[3] is typically used for preprocessing which is a way to extract the valuable information about a given image and used for input into a model or an algorithm.

Face Segmentation is where different facial components are separated into different colour layers[5].

Graphical effects are 2D or 3D effects overlaid on an image using the facial landmarks for aligning the graphics correctly.

We are assuming that the results will be sub-par but still roughly around the areas we want the points to be for the facial landmarks; additionally facial segmentation and the graphical effects is reliant on the landmarks so the effectiveness can only be based on them.

2 Methods

2.1 Dataset

I transformed the points dataset by dividing it by 1000 to get values between 0 and 1, then flattening each entry within the dataset. For the images I applied HOG to each image to train from. I created a 70%, 24%, 6% split with the data to be the training set, validation set and test set.

2.2 Preprocessing

Using preprocessing we can reduce the data to a smaller size but still keep the features intact to a certain degree, this reduces the training time meaning it will learn faster.

We used HOG; the HOG parameters we used, used nine orientations, five by five pixels per cell and one cell per block. This would return an array in the shape of (47, 47, 1, 1, 9) however was reshaped to be (47, 47, 9) as it made it easy for the two dimensional convolution to work. The function actually returns a feature vector, which would be a flattened edition of what the output is, but we can reshape it within the CNN to go back into shape.

2.3 CNN

Throughout the choices within the CNN I typically use a multiple of 92, which is the output size and will be reshaped to become the points to be used to locate the facial landmarks.

The CNN was a fairly basic linear model, Figure 1, written using tensorflow keras.

It first takes in the feature vector and reshapes it to be the original shape.

Next it is inputted into a two dimensional convolutional layer, this is for the initial feature pattern discovery using 368 filters, it reduces the size for the next layer.

After it is maxed pooled, and sent into another convolutional layer with 184 filters, further reducing the size. Now the array is flattened, and passed forward.

The dense layer has 644 units, we are using this layer and unit count because there are seven main clusters of points: left eye and eyebrow; right eye and eyebrow; the nose; mouth and the surrounding line of the face. 92 multiplied by 7 is 644 so it was chosen

The next layer is a dense layer, with 184 units, this is to reduce the size but as an intermediate between the actual coordinates.

Lastly, another dense layer to be size of 92, the output layer, which will be reshaped after to be the points array for the hog input.

All the layers, sans the last, use relu as the activation functions, this is because it overcomes the vanishing gradient problem which allows models to learn faster and perform better[1].

The last layer uses sigmoid because it makes the output numbers between 0 and 1 as this is the range of positions there are within the dataset.

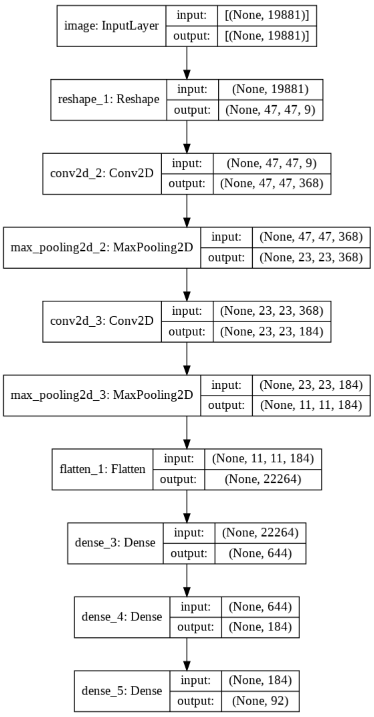

Figure 1: CNN Model.

2.4 Face Segmentation

We made a simple algorithm which reorganises the facial landmarks for a polygon to be drawn from each section which creates facial segmentation.

2.5 Graphical Effect

We made a simple googly eye effect by finding the midpoint of the two points of the eye to place the googly eye over, finding the length, multiplying the size of the googly eye by the length doubled. The eye is then placed over the image with an offset of the size of the length as the coordinates of the image start in the top left.

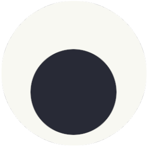

Figure 2: Googly eye effect.

3 Results

3.1 Facial-Alignment

The model has the accuracy of 86% on the testing data. We will discuss this later

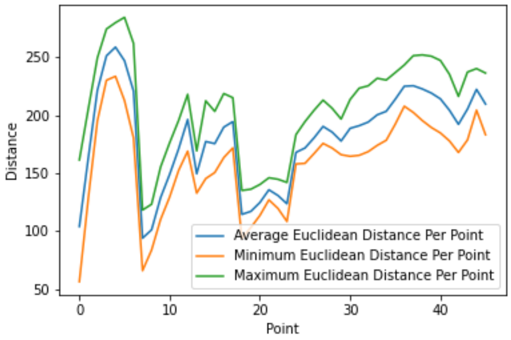

Figure 3: Euclidean distance per point showing average, minimum and maximum values.

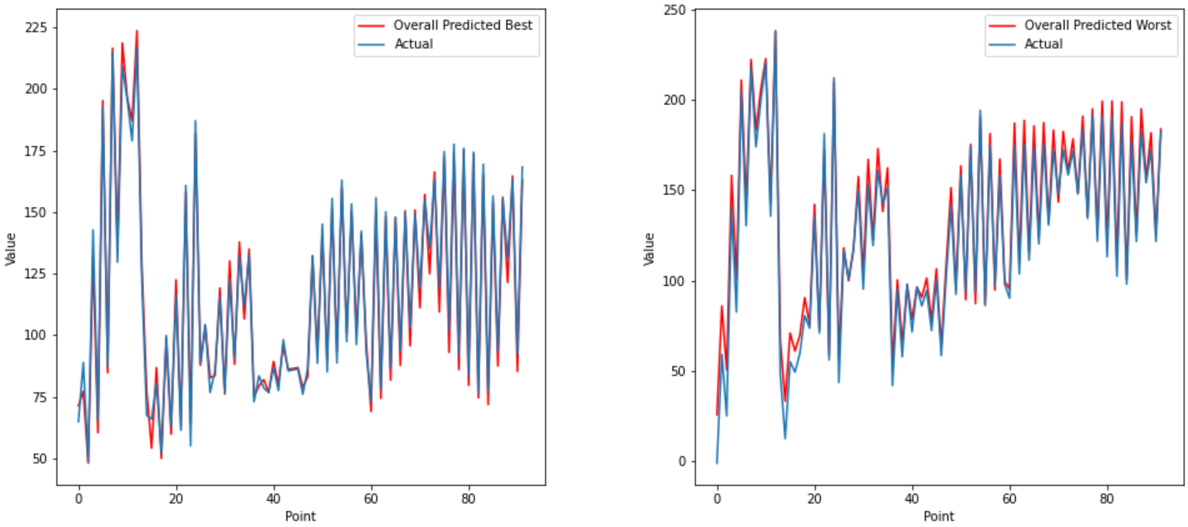

Figure 4: Best and worst results verses the actual points

In Figure 3, this shows each point is the euclidean distance between the predicted and actual points by three different categories. They all follow the same form, having most of the issues between 0 and 10 points and ≈25 to 42. The lack of difference between the different categories shows that it reliably scores 83%.

In Figure 4, the points array is flattened so there are 92 points now. In the overall predicted best, it doesn't reach the troughs as much as the peaks, but for the overall worst it doesn't reach the peaks but seems to more reliably hit the troughs.

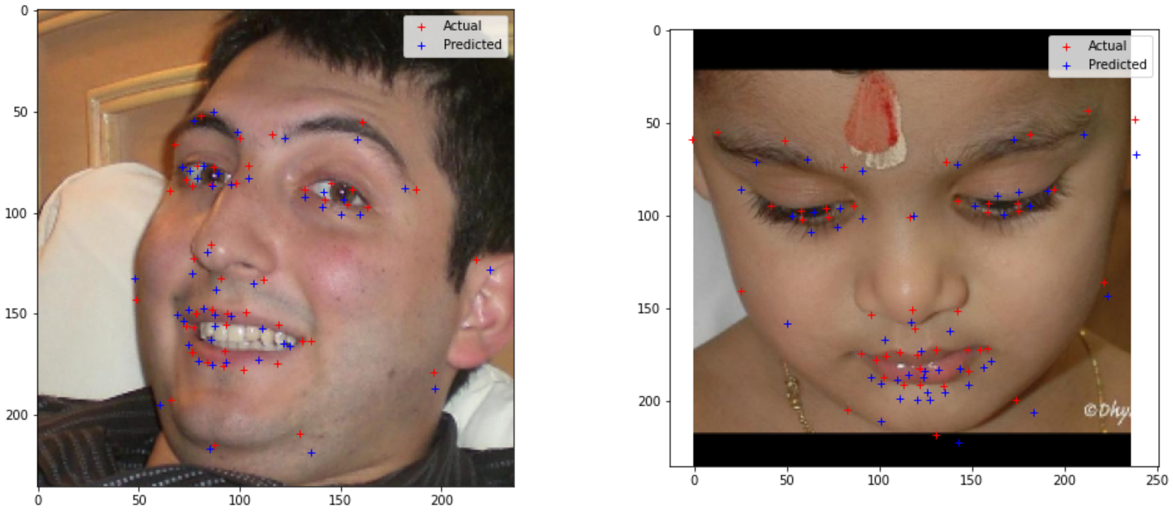

Figure 5: Left: Best result. Right: Worst result

Figure 5, using the best and worst euclidean distance results, the left image seems to be close for the majority of the points, however there is a large off-set with the mouth which would throw off the accuracy, while in the right, almost everything is wrong however are in the rough general area of where the facial landmarks should be however is still not accurate enough for real life implementation, but the angle of the image would not be as common as a full front face but you could also argue this for the left half too.

3.2 Face Segmentation

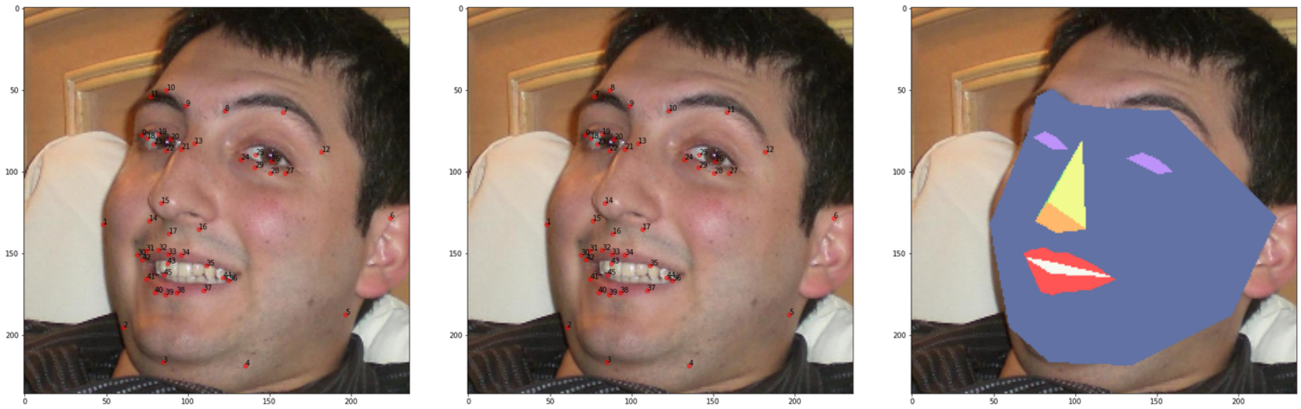

Figure 6: Facial segmentation flow chart

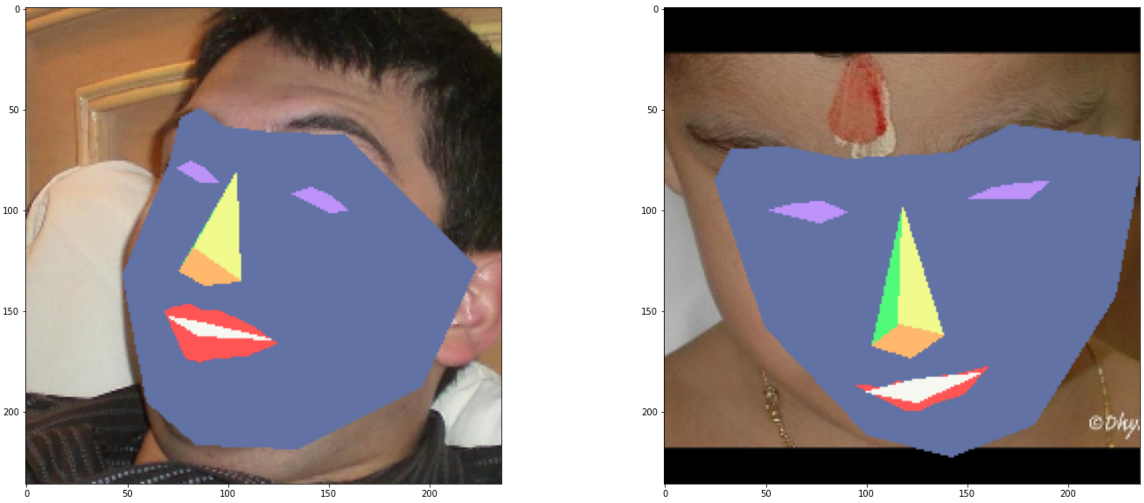

Figure 7: Left: Best result. Right: Worst result.

Figure 7, the facial segmentation is reliant on the results of the facial landmarks as it draws its results from there. On the left, it seems to have worked great with no visual glitches. On the right, it is perceived that the person is smiling with their teeth showing, but their teeth are seemingly outside of the boundary of their lips.

3.3 Graphical Effect

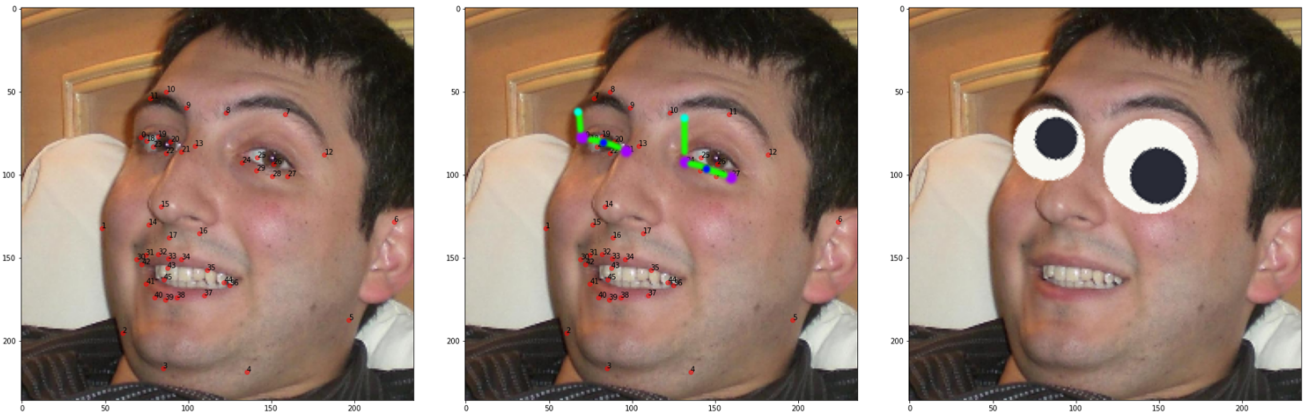

Figure 8: Graphical effect flow chart

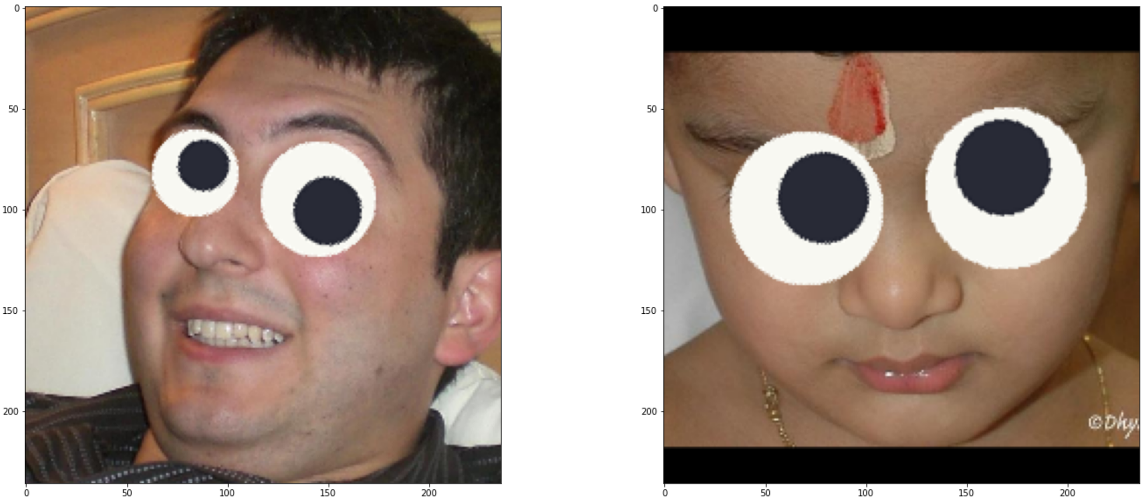

Figure 9: Left: Best result. Right: Worst result.

Figure 9, This is also reliant on the facial landmark accuracy, however more so around how accurate the eye corners are. If they had accurate placement where they were this would have no errors. The left image works really well, with the sizing from the length of the eye and the position working well. The right, due to the inaccuracies of the eyes, doesn't work well as it doesn't size or position well relative to the proportions and angle of the person's face.

4 Discussion

The accuracy of the model is good but not reliable enough for something to be implemented in a real-life / production system.

The model has learned the typical structure of faces, they can roughly place the markers around the correct areas of the face however the precision of those points being where they should be is visually low from the two examples given.

In Figure 4, it either hits the peaks or hits the troughs more accurately, this should be developed to try and create a better CNN model for this problem possibly having multiple convolutional 2D layers parallel which converge at some point which may give better results. Experimentation is needed as well as the theory itself.

Using HOG reduced the quantity of data to train with per image, to create a feature vector, however this reduces the graniality of the data which may make it too generalised in large areas where it becomes useless to train from.

Because of the nature of neural networks we do not know the full workings of the network, only the structure, which makes it more difficult to analyse than a hand crafted algorithm.

The dataset may affect the outcome due to the angle variety being disproportionately arranged and split into different clusters. I could mitigate this by shuffling all of the data rather than just shuffling the subsets next time.

Our method may not be the most optimal solution, another model could be using a Style Aggregated Network for Facial Landmark Detection[4].

The face segmentation relies on the points generated from the model so not much improvement can be made, this can also be said for the graphical effect, however the graphical effect could be improved by making a 3D graphical effect to make it more interesting however is out of the scope.

5 Conclusion

My model works well but not good enough for something in production. Facial segmentation and graphical effects are reliant on the accuracy of the model to work well. The model needs to be worked on or use a completely different method.

References

[1] Jason Brownlee. A gentle introduction to the rectified linear unit (relu), 2019.

[2] Adrian Bulat. *Deep learning for real world face alignment.* PhD thesis, 07 2019.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005.

[4] Xuanyi Dong, Yan Yan, Wanli Ouyang, and Yi Yang. Style aggregated network for facial landmark detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 379–388, 2018.

[5] Khalil Khan, Rehan Ullah Khan, Kashif Ahmad, Farman Ali, and Kyung-Sup Kwak. Face segmentation: A journey from classical to deep learning paradigm, approaches, trends, and directions. *IEEE Access*, 8:58683–58699, 2020.

[6] Thomas Wood. Convolutional neural network, 2020.