Locating Facial Landmarks in images using Ridge Regression

Candidate 246518

Computer Vision (G6032)

May 11, 2023

1 Introduction

Dense facial landmark detection is a key element of many computer vision and biometric systems, providing essential spatial information for applications such as facial recognition, emotion analysis and facial animation. (Oloyede, 2020).

Early approaches of facial landmark detection were primarily designed for use within controlled environments, where lighting was optimal, facial pose was clear and occlusion was minimal, aspects which are challenging for computer vision systems to handle.
These early systems often struggled 'in the wild' due to the presence of these challenges. (Khabarlak, 2021)

Within this report, I aim to create a simple machine learning algorithm similar to these early approaches, to detect a set of 44 key landmarks on a face, the locations of which can be found below.

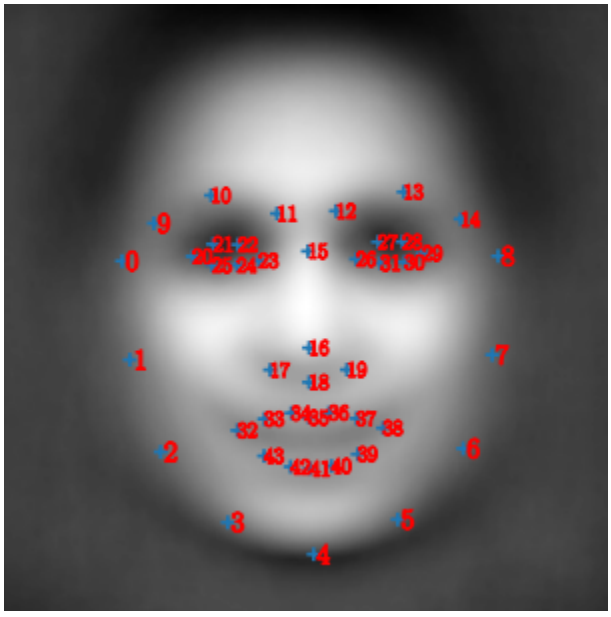

Figure 1: Illustration of the locations of the 44 points on a face.

I aim to optimise this through testing a variety of different parameters and hyperparameters for optimal landmark detection using pre-processing techniques, HOG gradient detectors and through training a ridge regressor. An attempt will also be made to improve this algorithm using a 5-point data subset. The

performance of these will be measured using the average mean Euclidean distance between points within the image over a labelled validation dataset.

2 Method

2.1 Data Pre-processing

Several Pre-processing steps have been applied to each image within the dataset to make it more suitable for face detection. The steps and justifications of which are as follows:

Image resizing

To start, I experimented with resizing the images to a smaller scale to reduce the amount of data that further steps would have to use in the hope that performance would increase due to noisy information being lost, improving edge/gradient detection.

Testing these predictions using image sizes of 256x256, 128x128, 96x96, 64x64, 48x48, 32x32 and 16x16 pixels yielded the following results:

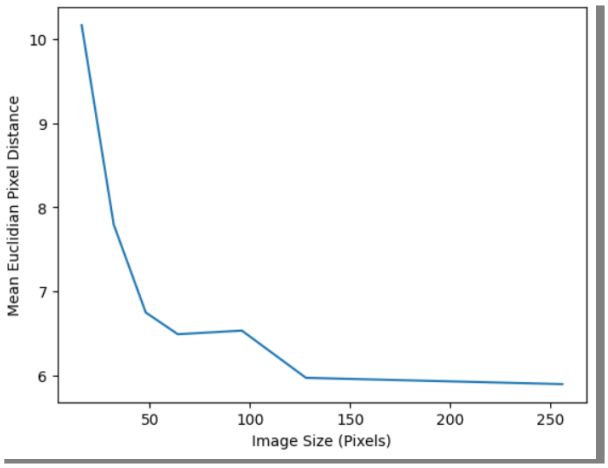

Figure 2: Mean Euclidean distance plotted against different image sizes.

Contrary to my expectations, larger image sizes gave higher performance values, with the best performing image size being 256x256, with a mean Euclidean distance of 5.89 Pixels (3SF).

Due to this, for optimal results on the testing data, no image resizing will be done.

(It is worth noting that the Gaussian blur kernel size & HOG window sizes were adjusted to scale with the different image sizes, and that all predictions were resized to 256x256 when calculating mean Euclidean distance.)

Greyscaling

Each image's colour information is removed by applying a greyscale filter. Greyscaling the data helps remove any colour information, which may be helpful in reducing representational differences between skin tones, colour temperatures and lighting colours.

Greyscaling also helps to simplify the task by reducing the size of our training data by a factor of 3, as the red, green and blue channels are being averaged, reducing an image's dimensionality from 3 dimensions to 2 dimensions.

Histogram equalisation

After being greyscaled, each image undergoes histogram equalisation to increase the contrast between edges, making the edges within faces clearer for the edge detector of the predictor. (Yepeng Liu, 2021)

Gaussian Blurring

Gaussian blurring was another pre-processing step which was tested for this task, as a different form of noise removal (opposed to downscaling). Gaussian Kernel sizes of 1 (no blurring), 3, 5, 7, 9, 15, 31 and 51 pixels were tested, with my prediction being that a small kernel size of around 3-7 would improve performance by removing noise, while keeping the edges on a person's face distinct.

The experiment yielded the following results:

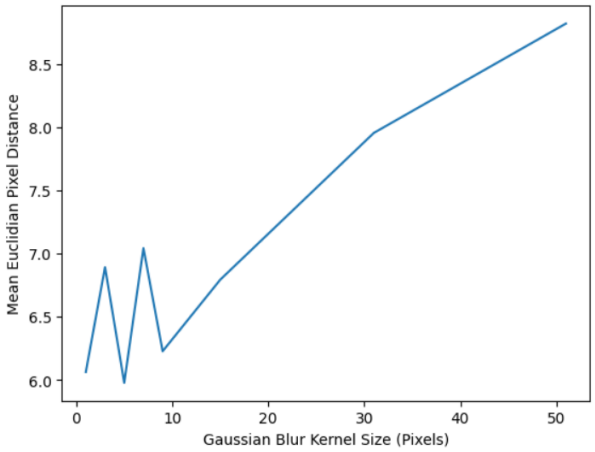

Figure 3: Mean Euclidean distance plotted against different Gaussian Blur kernel sizes.

While the data between kernel sizes 1-10 is somewhat erratic, there is an overall trend of performance degrading as the gaussian kernel gets larger, starting the 9x9 kernel.

The best performing kernel size was 5 pixels, with a mean Euclidean distance of 5.98 Pixels. Due to this, for optimal results on the testing data, a gaussian filter with a kernel size of 5 pixels will be applied.

2.2 HOG gradient detection

After blurring, a Histogram of Oriented Gradients (HOG) is calculated from the image. HOGs capture the distribution of gradients over several "cells" within an image, converting it from a standard image to a list of

histograms. This is done by computing the gradient magnitude and orientation for each pixel and dividing the image into small cells. (Singh, 2019)

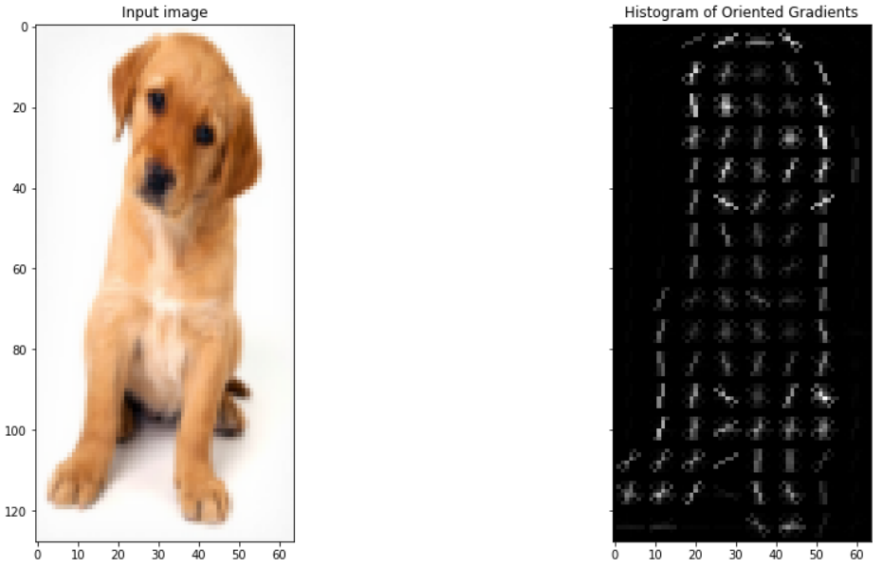

Figure 4: The HOG of an image of a puppy (Singh, 2019)

The HOG Parameters tested are as follows:

- The size of each cell within the image in pixels (set as a square for the sake of reduced complexity).
- The Number of Gradient bins that the detector has.

With the previously calculated pre-processing parameters, I experimented using different cell sizes (64,48,32,16,8) and bin quantities (4,6,8,10,12,16,64), with the following results:

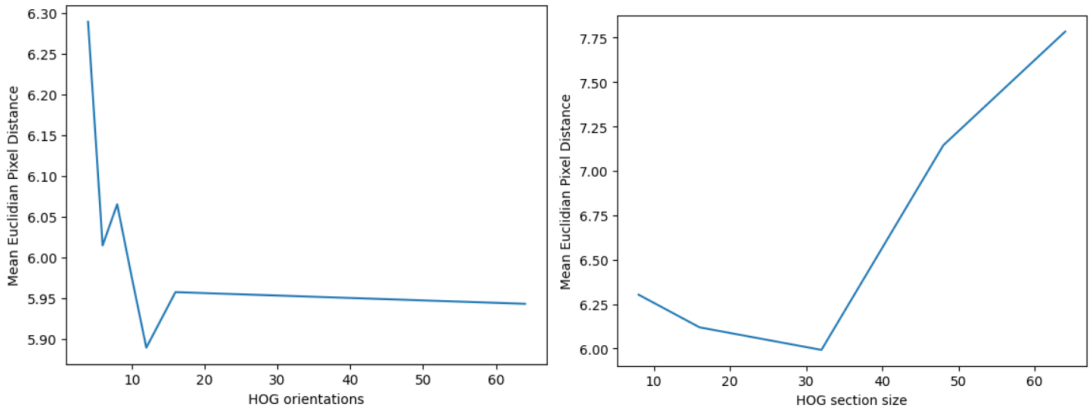

Figure 5a, 5b: Mean Euclidean distance plotted against different numbers of Gradient Bins (Left) Mean Euclidean distance plotted against different HOG Cell Sizes (Right)

As the graphs above show, the optimal HOG parameters are to use a 32x32 cell size with 12 gradient orientations.

Pre-processing time was massively affected by decreasing the HOG's cell sizes, as more calculations were necessary to create a larger number of cells within each image, taking around 30 minutes for a cell size of 4, which was excluded from the experiment above for taking too long.

2.3 Ridge Regressor training

I chose to use a ridge regressor (RR) to predict the face landmarks primarily due to its ability to reduce overfitting through its regularisation system.

As faces have a certain level of intra-class variation (skin tones, face shape, poses) it is important that the RR can generalise to new, unseen faces which vary from the training data.

I have found that RRs are also faster to train and easier to use than other approaches due to their simplistic nature (in comparison to modern CNNs and DNNs). Which allows for easier & faster experimentation with parameters.

The primary hyperparameter to consider when using an RR is its alpha value. This alpha value is a penalty coefficient which can be tuned to allow or discourage overfitting of the training data provided.

Using the optimal pre-processing steps discovered before, I trained and evaluated several RRs with alpha values of 1, 10, 25, 50, 100, 200, 300 and 500. The results are as follows:

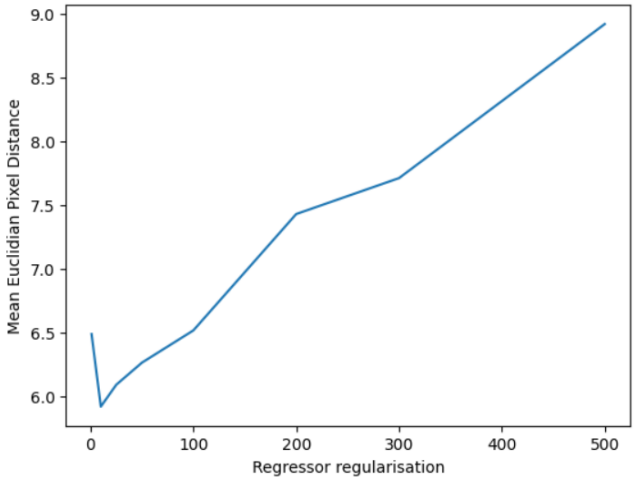

Figure 6: Mean Euclidean distance plotted against different RR Alpha Values.

The above graph shows a mostly positive correlation between mean Euclidian distance and Alpha Values, except for values below 10. Finding the optimal RR Alpha value as 10 with a Mean Euclidean distance of 5.92 pixels.

2.4 Predicting

With all these optimised parameters, the procedure for training and predicting for the detector are as follows:

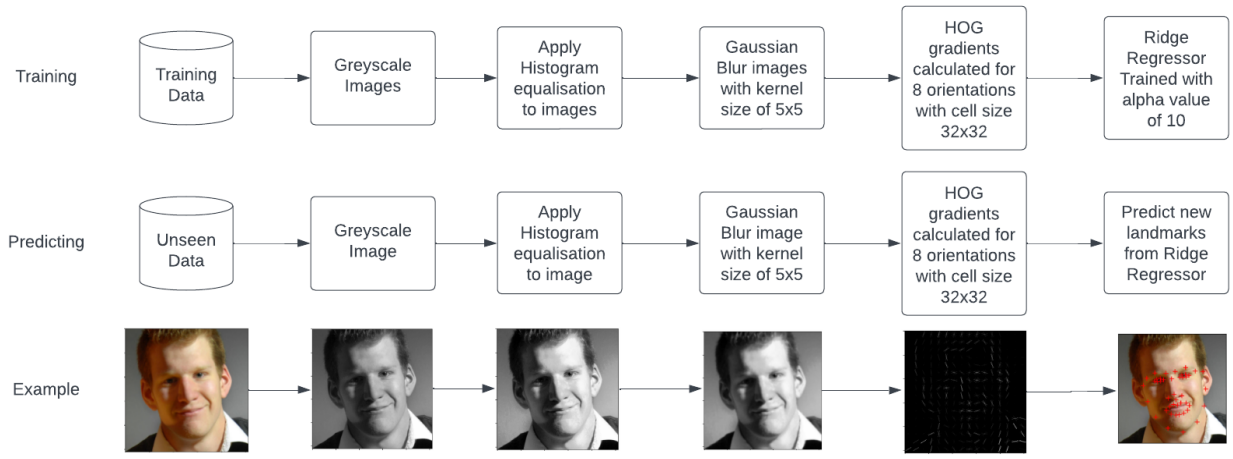

Figure 7: The Training and predicting process for the detector, with an example image.

2.5 Alternative approach with subset points

An attempt to further improve the detector was made by using the 5 subset landmarks to predict the other 44 landmarks. A first RR is trained to predict the 5 subset points from a pre-processed image. These 5 points are then fed into a second RR which predicts the other 44 points from the 5 points. A flowchart of how this is done can be seen below.

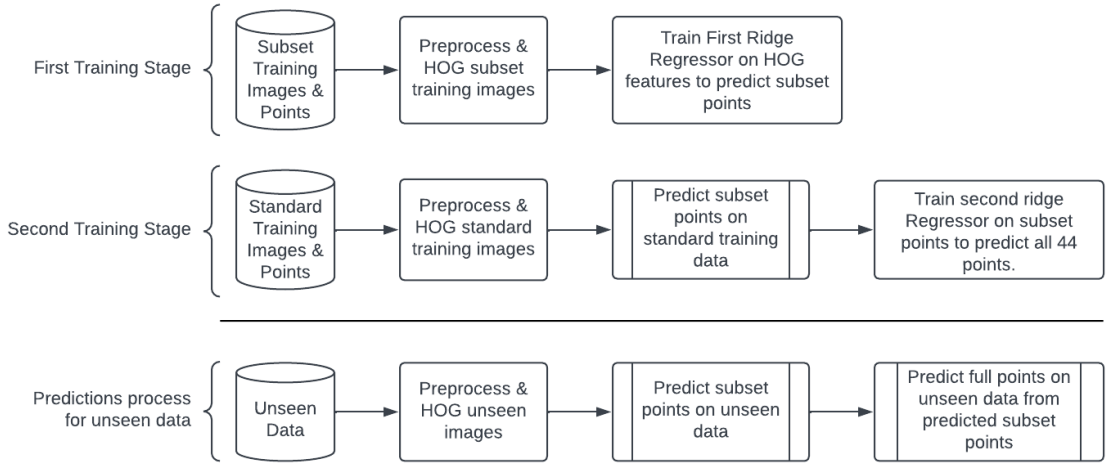

Figure 8: Alternative approach Training & Prediction Process

3 Results

Testing of the subset and standard detectors on a validation set shown that the standard detector was slightly better with a final mean Euclidean distance of 5.88 pixels, as opposed to 5.96 pixels, both of which were calculated after training and running each detector 5 times and averaging the results.

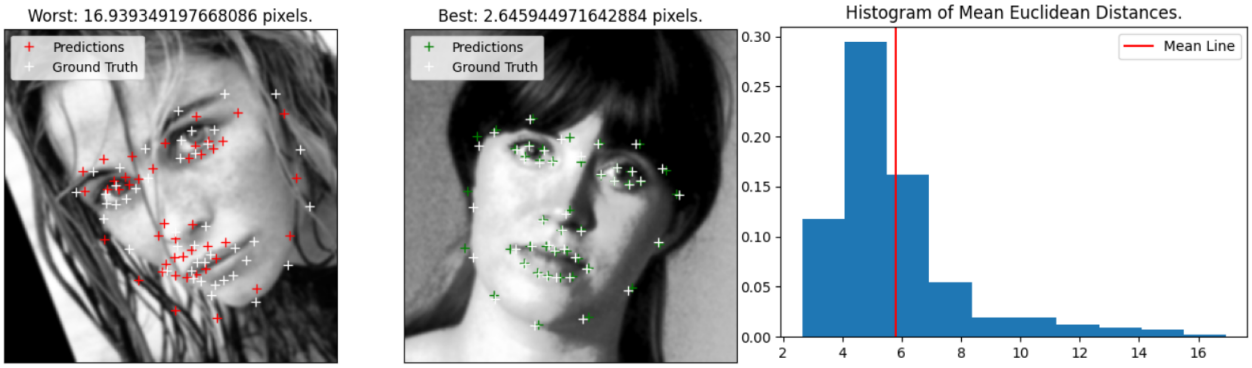

Figure 9: The best & worst predictions using the standard detector, with its mean Euclidean distance Histogram over the validation set.

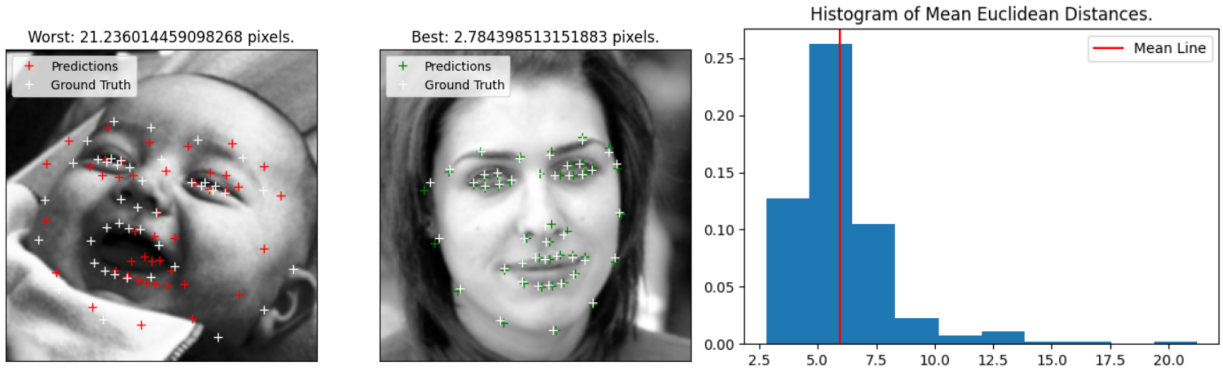

Figure 10: The best & worst predictions using the alternative detector, with its mean Euclidean distance Histogram over the validation set.

While not perfect, I am surprised by the performance of the system and its ability to identify facial features on unseen data, considering its simplicity.

The Standard detector will be used due to its lower Average Mean Euclidean Distance.

4 Discussion

4.1 Parameter optimisation shortcomings

One concern I have is the possibility that the detector has fallen into a local minimum from the random starting parameters I gave the detector.

Combining more processing power with a Genetic Algorithm may have been more suitable than the manual process I carried out, using a sparely distributed population of different parameters (genes) to avoid converging on a single local minimum, with one gene hopefully falling into the global minimum.

4.2 Failures & unexpected behaviour

Looking through the example set predictions, some unusual behaviours can be spotted.

- The Algorithm often predicts the top of the eyes to be slightly lower than where they are, which I believe is caused by the gradient of the eye's pupils and whites being used by the classifier, instead of the shadow created by the eyelid.

Figure 11: An example of the eyes being incorrectly detected.

- The algorithm seems to struggle with mouths, especially closed ones, typically predicting the point where the two lips meet to the boundary of the top or bottom lip.

  I believe this behaviour is caused by the whole mouth being generalised into one HOG cell. As many mouths in the dataset are open, the predictor attempts to generalise towards open mouths and places the lips slightly further apart than they typically would with a closed mouth.

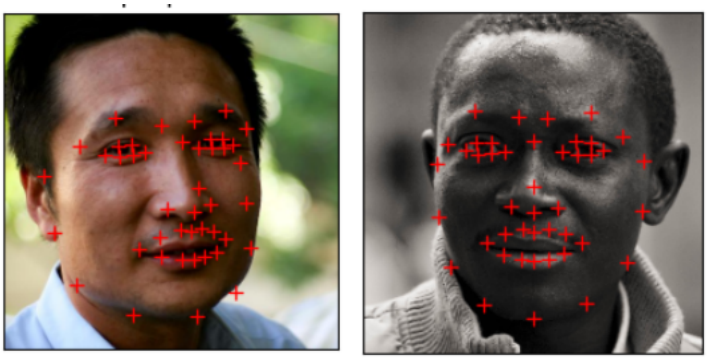

Figure 12: Examples of the lips being incorrectly detected.

4 Conclusion

To conclude, this report has detailed the development and optimisation of a simple machine learning algorithm to detect 44 facial landmarks on faces using pre-processing steps, HOG gradient detectors with a trained ridge regressor, as well as an attempt to use a 5-point subset to increase performance.

5 Bibliography

Khabarlak, K. &. (2021). Fast facial landmark detection and applications: A survey. *arXiv preprint arXiv:2101.10808*.

Oloyede, M. H. (2020). *A review on face recognition systems: recent approaches and challenges.* Ilorin: Springer.

Singh, A. (2019, September 4). *Feature Engineering for Images: A Valuable Introduction to the HOG Feature Descriptor*. Retrieved from Analytics Vidhya:

https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/

Stefanos Zafeiriou, C. Z. (2015). A survey on face detection in the wild: Past, present and future. *Computer Vision and Image Understanding*.

Yepeng Liu, F. Z. (2021). Image smoothing based on histogram equalized content-aware patches and direction-constrained sparse gradients. *Signal Processing*.