An approach to locate facial landmarks using a Convolution Neural Network

Candidate Number: 246619

May 11, 2023

1 Introduction

To perform the task of facial landmark detection I decided to use a Convolutional Neural Network (ConvNet). They excel at finding hierarchical features from raw data [1], starting from those at the low-level (corners and edges), combining these into progressively more advanced high-level features [2] such as noses, ears and mouths. At the cutting edge of the field tweaked Heatmap based ConvNets are used [3], but a simple direct regression model can be trained to reliably detect facial landmarks. The benefits to this approach, over others such as ridge regression, is that minimal pre-processing is required of the input images.

2 Methods

The black box nature of Neural Networks make training a model for a specific task inherently tricky. I experimented with many different types of architectures and hyper-parameters while constructing a facial alignment system with reasonable accuracy. I found the most success with the model outlined in Figure 1.

The loss function used was the mean squared error and to optimise the model I used Stochastic Gradient Descent. I tried many different sized mini-batches when training and had the most success with a batch size of 1, effectively updating the model parameters after each inputted image.

With early experimentation, I found that my loss function would increase to infinity after only a few epochs as the model suffered from exploding gradients. Batch normalisation applied after each layer solved this issue by smoothing the optimisation landscape and gradient build up.

While experimenting I attempted applying transfer learning to the task by using the pretrained model ResNet-18 [4]. I modified both the input and output layer to apply it to the facial alignment task, but ultimately found performance worse that the ConvNet previously outlined.

The dataset provided was split into a training, validation and test set - with the training set being 80% of the full data. Both the training and validation set were shuffled before each training epoch to increase the models ability to generalise and to reduce over-fitting.

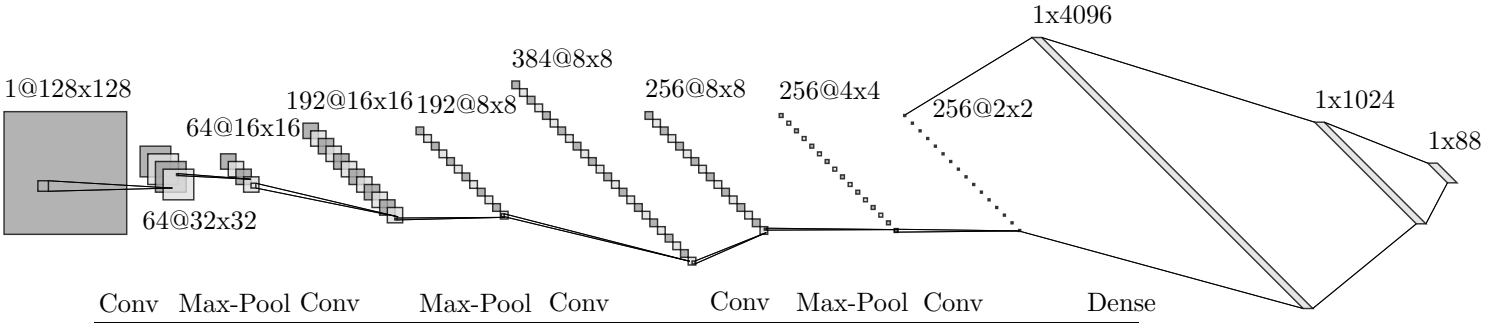

Figure 1: Architecture of the Convolutional Neural Network used

2.1 Pre-processing

Despite the powerful nature of a ConvNet, some pre-processing to the training data was required to achieve accurate results and a faster convergence when training. Figure 2 shows the entire pre-processing stage. First the image was converted to greyscale and downscaled to half the original resolution. The loss of detail minimally affected the prediction task, but greatly reduced the model complexity. Next the image is ran through a ColourJitter function which randomly alters the brightness and contrast, followed by a random rotation applied between +/- 10°. Both of these transformations aim to artificially extend the dataset as they are applied with different parameters at each epoch. I found that they improve the models generalisation, while also combating over-fitting.

Finally, both the images and the landmarks were normalised. This I found to be an essential pre-processing step. The features were transformed into a common range so that greater numeric features did not dominate the lesser features. The presence of greater features can negatively effect the capacity of the network to learn and results in a model that struggles to converge [5]. I found the best performance once I had normalised the training data landmarks between 0 and 1, but then centering the data around 0 by subtracting 0.5. Before completing normalisation, the model had a tendency to completely under-fit to the data and would predict the same mean face shape for all images in the test set as shown in Figure 3. I believe this to be evidence that the convolution layers had not learnt any of the features from the training data and instead the fully connected layers had approximated the mean face shape to minimise the loss function.

Figure 2: Pre-processing pipeline

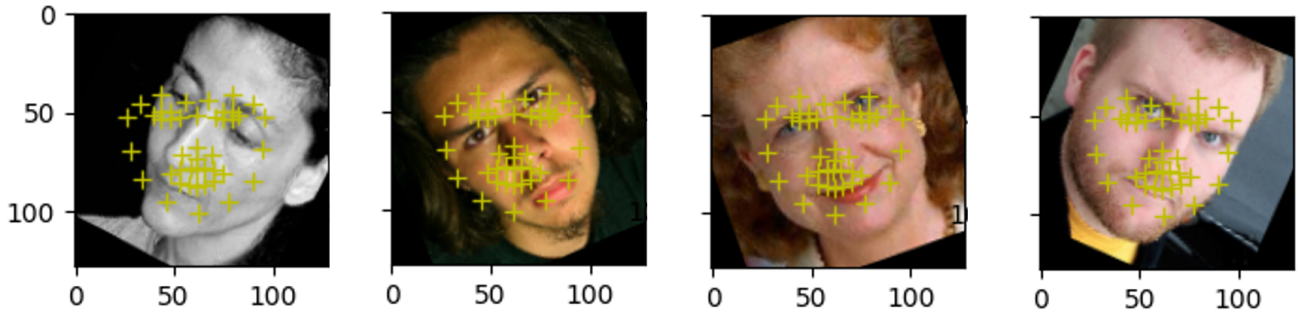

Figure 3: Evidence of the model under-fitting before applying normalisation to the inputs

2.2 Secondary Task

In order to use the training data with only 5 landmarks, I aimed to use the 5 points to predict the full 44 landmarks for each image. With the full set of landmarks, it would then be possible to append this data to the original dataset. The full workflow is demonstrated by Figure 4.

To begin, I extracted the subset of 5 landmarks from the full training set. Using the set of 5 and the full 44 landmark coordinates, I trained an Sklearn linear regression model to predict the location of the full 44 from the 5 points. Holding back 20% of the full training images I found that the linear regression model achieved a decent mean absolute error of 2.47 on a test set. As you can see in Figure 5, this proved relatively accurate with a mean euclidean distance between the ground truth points and the predicted points as 3.27 pixels across the test set.

Using the trained linear model, I then predicted the 44 landmarks for the data where only the 5 were provided. Concatenating this with the original dataset resulted in a set of 2811 images consisting of a mix of 1425 ground truth landmarks and 1386 predicted landmarks. I then trained the same ConvNet as had been used previously with the extended dataset.

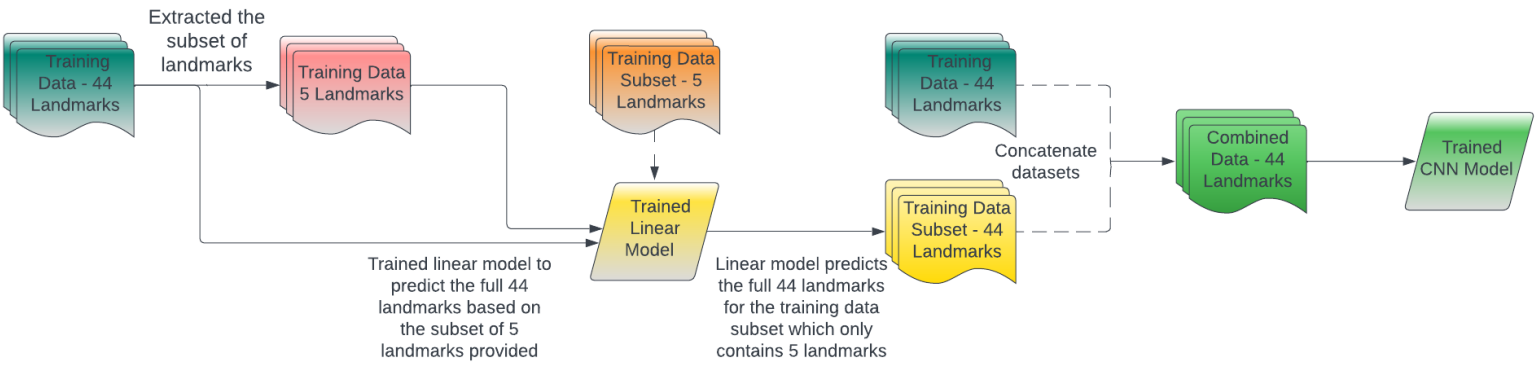

Figure 4: Flow chart of the methods used to incorporate the training data with a subset of landmark points into the prediction model

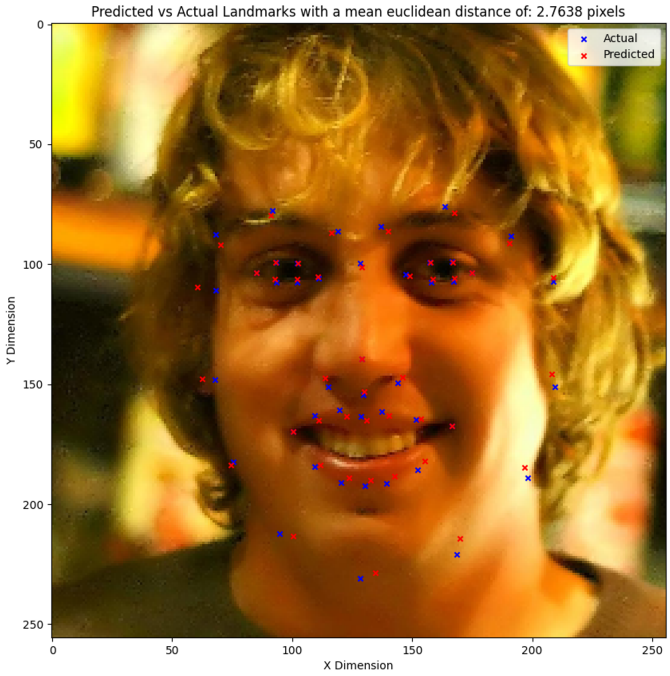

Figure 5: An example of the predicted 44 landmarks from a subset of 5 landmarks, compared to the 44 ground truth landmarks

3 Results & Discussion

Trained on the original dataset, the ConvNet performed well, resulting in a mean euclidean distance between the predicted and ground truth landmarks of 3.49 pixels across the test set. As seen in Figure 6, there were a few outlier images which had a mean euclidean distance of around 9 pixels, but overall a normal distribution was achieved. Training loss continued to decrease, whereas validation loss tended to slightly increase from the 12^{th} epoch. I believe this is a sign that the model is starting to overfit to the training data, as an extension to this report experimenting with dropout layers would potentially solve this shortcoming [6].

The model worked very well on images where the face was displaying a neutral expression and orientated straight on as seen in Figure 8. Systemic poor accuracy was achieved when either the face shape was uncommon, such as a babies (Figure 8), or the facial expression was extreme (Figure 8). I believe this failing can be attributed to the lack of images in the dataset which display such features, as the set contains predominantly adults with neutral/smiling expressions. I believe a more varied dataset would improve the models ability to generalise and increase the accuracy for examples such as these.

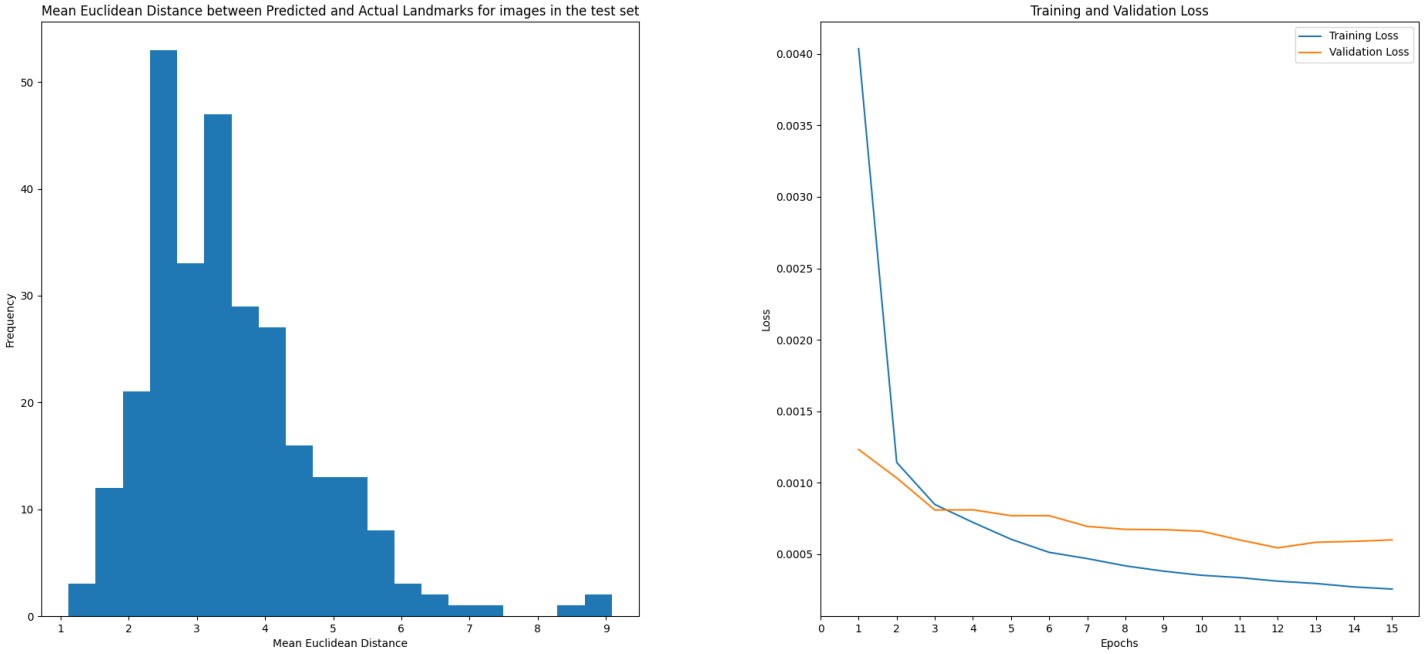

Figure 6: Performance metrics of the model averaged over 5 runs

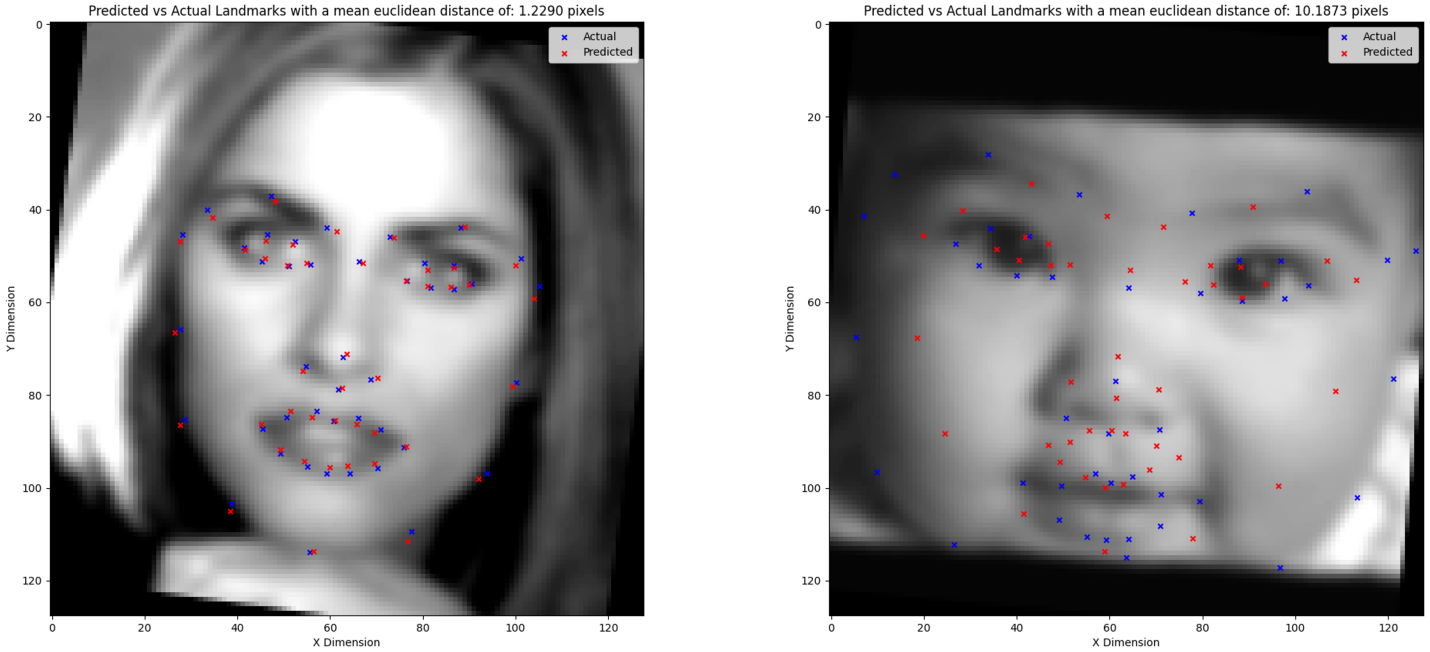

Figure 7: Example of the best and worst accuracy predictions found in the test set

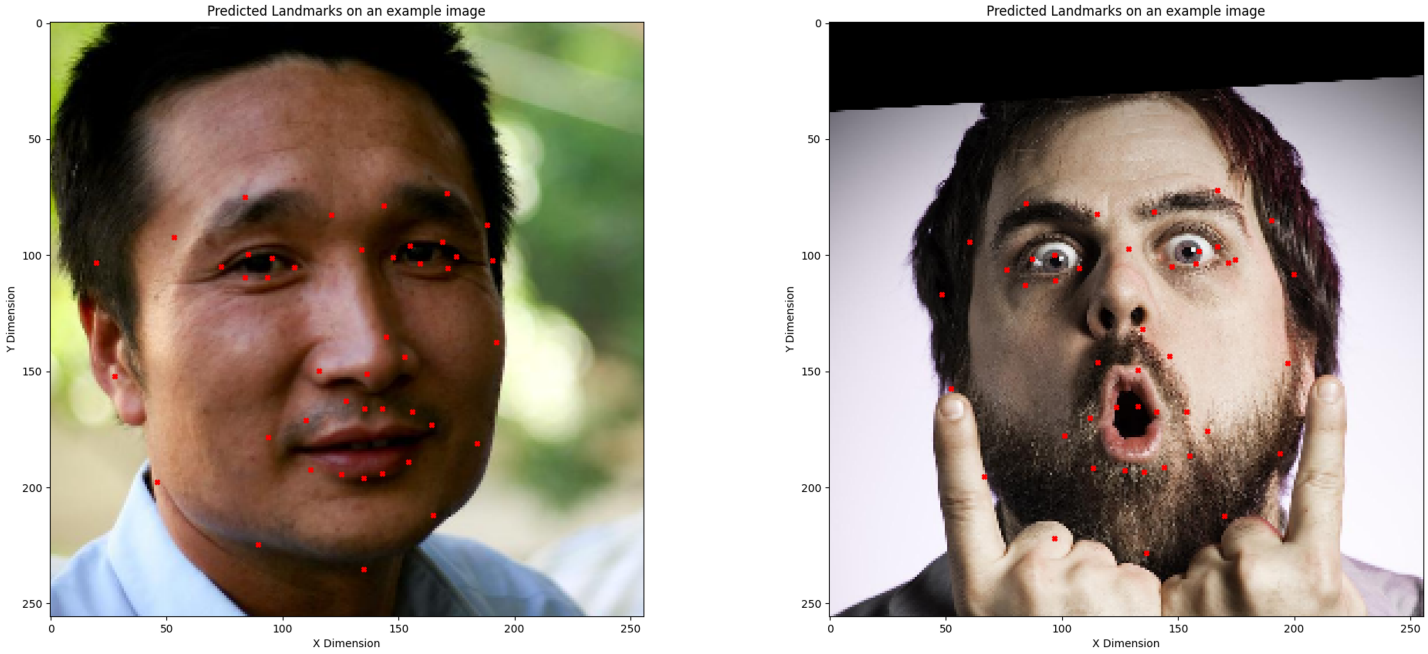

Figure 8: Predicted landmarks for images in the example set

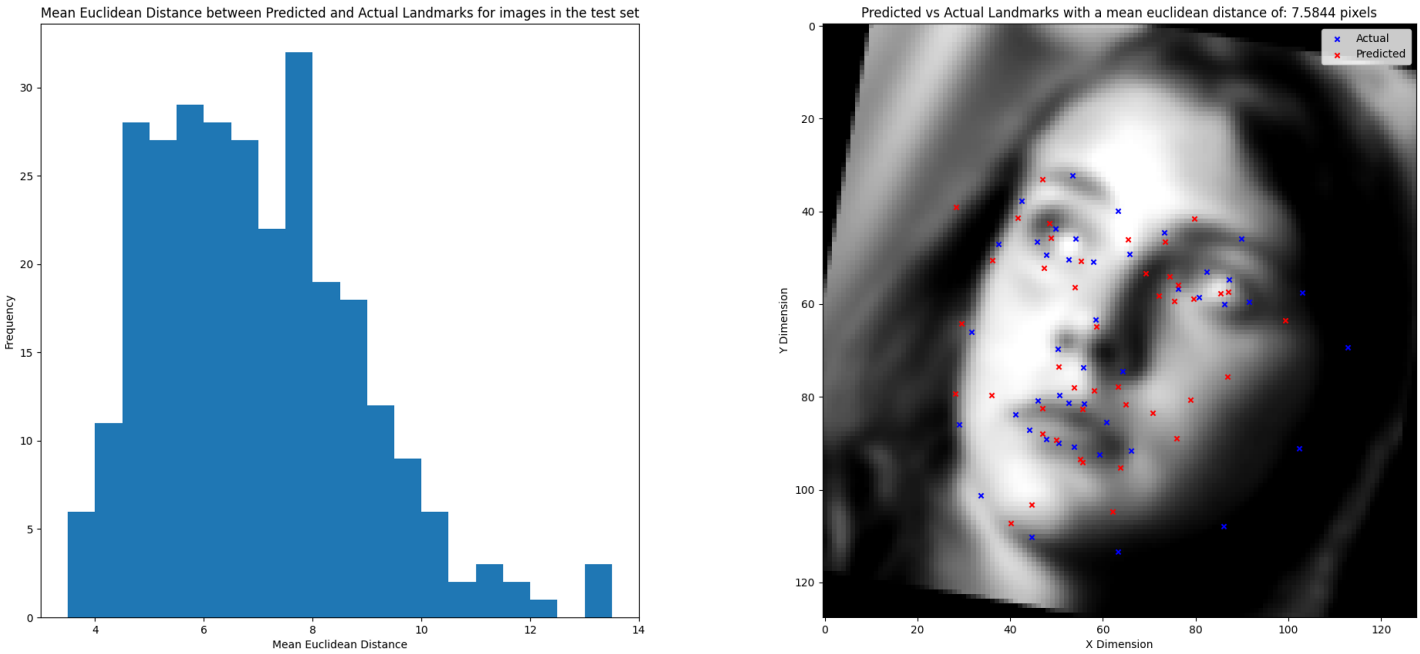

Figure 9: Accuracy of the ResNet-18 model and an example of one of the test images

3.1 Performance compared to ResNet-18

In comparison when using the pre-trained ResNet-18, the model struggled to locate any facial landmarks reliably. The model also failed to pick up the structure of a face in my experimentation, as seen in Figure 9.

3.2 Secondary Task

When training the model using the extended dataset, I withheld 20% of the images from the original set so that I would have a test set of 285 images of which the 44 landmarks were their ground truth coordinates. I did this so that I would be able to fairly compare the two models, rather than the 'ground truth' coordinates being those that had been predicted by linear regression as outlined in 2.2. The extended dataset benefited the model greatly. Averaged across 5 runs, the mean euclidean distance between the predicted and ground truth landmarks was 3.06 pixels - an improvement of 0.43 pixels on the original model. When comparing the example images in Figure 11 to those produced by the original model, the predictions are much more accurate around the mouth and eyes. Furthermore the predictions are not as likely to construct generalised features where is not appropriate. As seen in Figure 10, the mean euclidean distances between the predicted and ground truth landmarks for images in the test set have a tighter grouping and outliers are not as prevalent as in the original model. These metrics show that the variance of the model has increased as it has been trained on larger dataset, despite just under half of the training data not having hand placed landmarks.

Despite the higher accuracy achieved, the model still performed poorly on exaggerated facial expressions as seen in Figure 11. The additional dataset contained images of faces displaying similar expressions to those found in the original dataset, so it did little to help train the model for any of the previous outliers found.

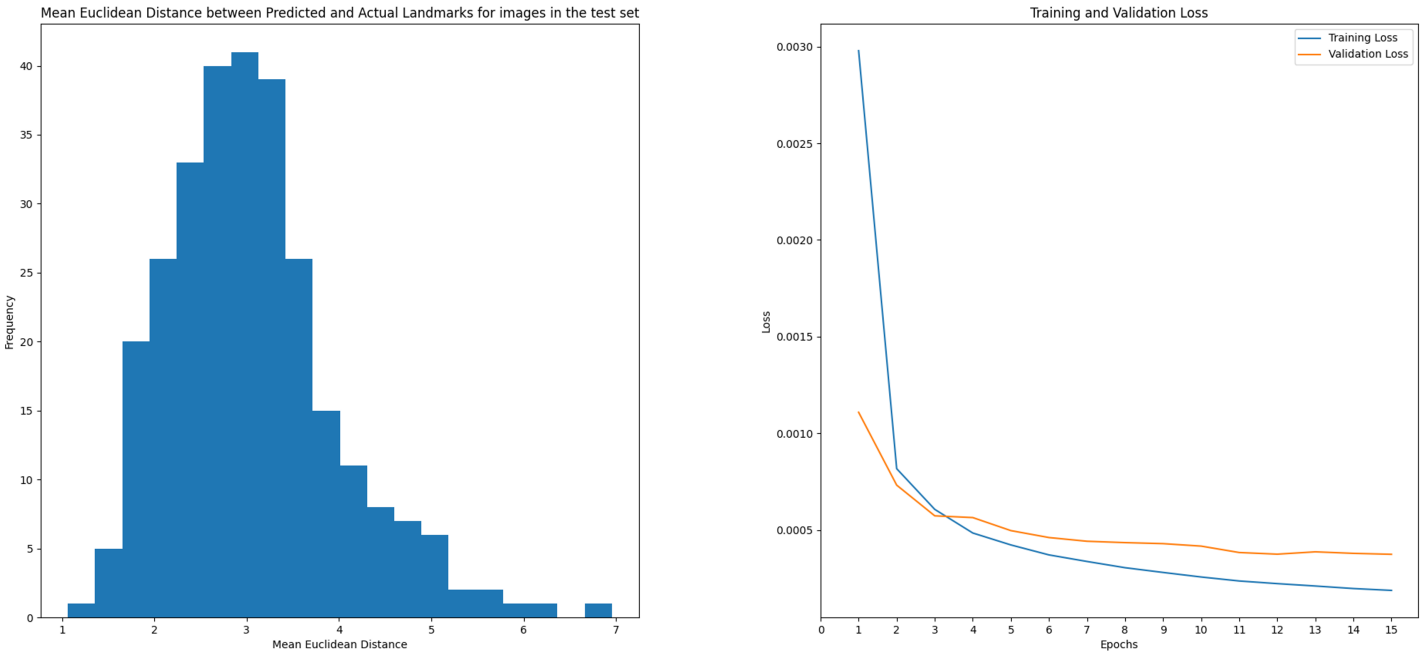

Figure 10: Performance metrics of the extended dataset model averaged over 5 runs

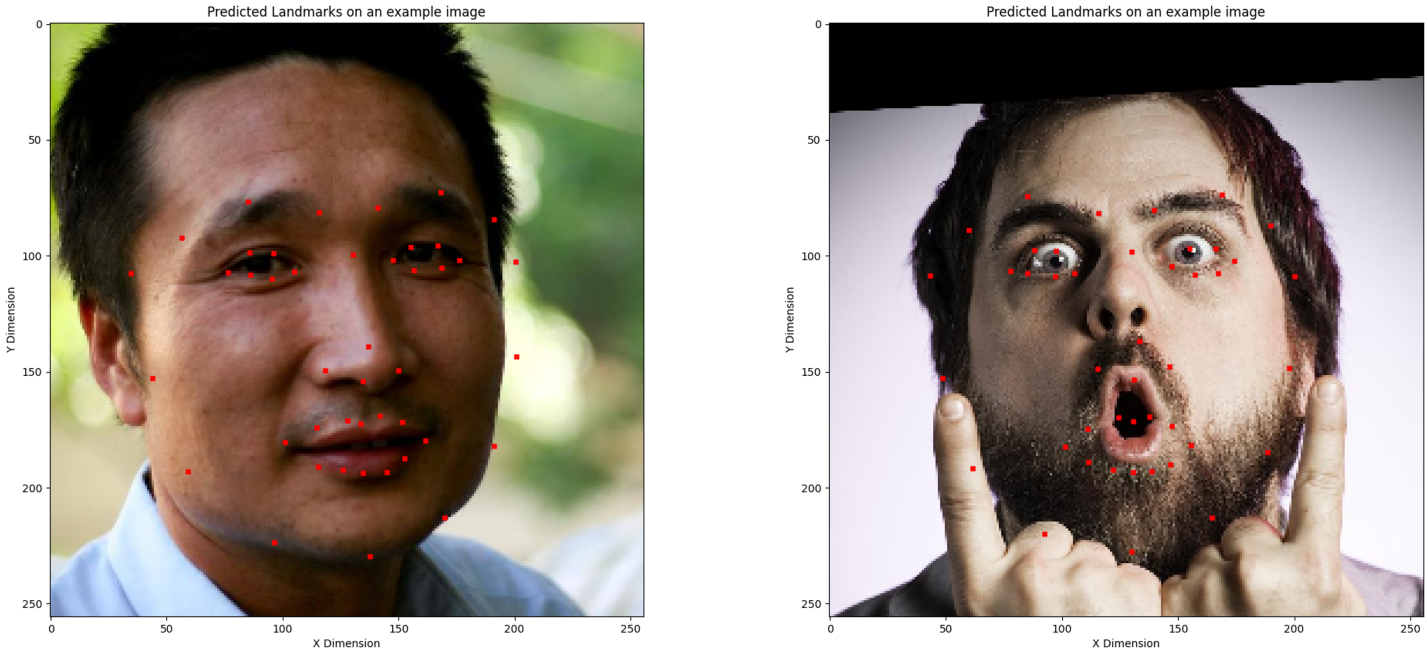

Figure 11: Predicted landmarks by the extended model for images in the example set

4 Conclusion

In conclusion, this paper presented a ConvNet approach to solving the task of facial landmark detection. Overall I believe I have achieved accurate results and by further processing the extra data with a subset of landmarks, futher improved the models generalisation.

As discussed in Hsu et al [3], the direct regression approach, while good at maintaining a structure of a face, it does so at the cost of slightly inaccurate landmark positions. Given the relative uniformity in the dataset, the method I have chosen works fairly well but for future work if given a more varied dataset or real world use case; I would attempt a Heatmap approach.

References

[1] Yue Wu and Tal Hassner. Facial landmark detection with tweaked convolutional neural networks. *CoRR*, abs/1511.04031, 2015.

[2] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9(4):611–629, Aug 2018.

[3] Chih-Fan Hsu, Chia-Ching Lin, Ting-Yang Hung, Chin-Laung Lei, and Kuan-Ta Chen. A detailed look at cnn-based approaches in facial landmark detection. *CoRR*, abs/2005.08649, 2020.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[5] Dalwinder Singh and Birmohan Singh. Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, 97:105524, 2020.

[6] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.