Student Number: 183708

Abstract

Facial Landmark Detection (FLD) is a task of detecting key landmarks in the facial area. This paper focuses on predicting facial landmarks in unseen images using a convolutional neural network (CNN). The purpose of this paper is to surmise and show a CNN type approach to finding a solution to landmark detection, rather than present a novel idea or method. From there we can highlight the benefits of such an approach and discuss possible shortcomings.

***Keywords*—** Facial Landmark Detection, Convolutional Neural Network

1. Introduction

This paper uses a dataset of labelled sample images, labels come in the form of sixty-eight points each represented by two separate values: x and y coordinates. Points are grouped in an ascending order depending on the area of the face they map, as seen in Figure 1. This is a supervised learning system where the labelled image data is fed into the neural network which will then predict sixty-eight landmarks per image throughout a test set. Using metrics such as Euclidean Distance and mean squared error (mse) we can compare different CNN structures and their effects on performance during training.

2. Deciding Approach

Popular approaches to FLD include template fitting [2, 12] or regression-based face alignment algorithms [1, 13]. Cascaded regression trees and cascaded deep CNNs have been shown to provide great results [14] too. The former being able to perform accurate facial alignment in a manner of seconds [8], whilst the latter took uses two triumphs of computer science (Cascaded Regression and CNNs) to create the Cascaded CNN [17]. However, these methods use complex cascading architectures and considering the small size of the problem-set it was decided that a simple CNN would suffice in predicting facial landmark points, subject to computationally limited resources. Pre-processing data is also less necessary in CNNs when compared to other similar algorithms. With enough training a CNN is able to learn the characteristics of its training data, and unlike a MLP, it is able to understand complex spatial dependencies within an image.

2.1. The Convolutional Layer

By applying a filter across a multi-dimensional array, a map of activations 2 is created which indicates the strength of detected features in an input and that feature's location. In mathematics this is known as cross-correlation rather than convolution and is used in many machine learning libraries [4]. Depending on padding around the input, it can be a tool for dimensionality control [3].

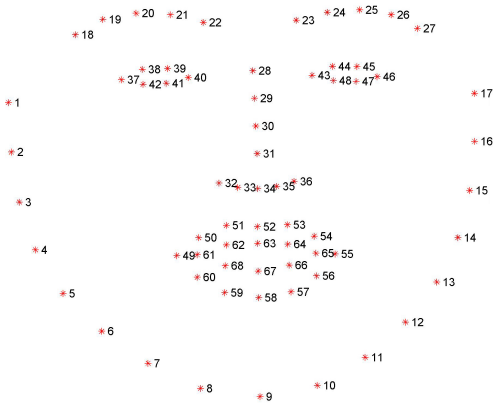

Figure 1. IBUG 68 Landmark Points

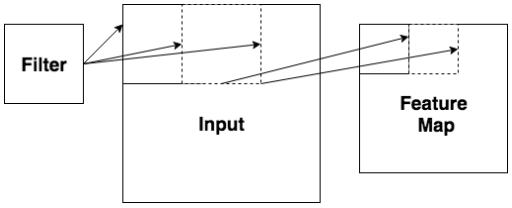

Figure 2. Overlapping of a Filter in Convolutional Layer

2.2. The Pooling Layer

There are two main choices for pooling: Average pooling - average of a kernel over portions of an image, and max-pooling - maximum value of a kernel. Often max-pooling is compared to dropout as it can help a model avoid over-fitting by acting as a regularizer [5], although this is more the case with overlapping pooling [10]. Its main purpose here is to suppress what little noise exists and reduce dimensionality of the data.

2.3. The Dense Layer

A fully connected neural network layer. Each neuron in the layer has input from the previous layers neuron and are all connected to those in the next layer. These layers are often placed at the top (last layer) of model to perform the prediction task.

3. Convolutional Neural Network Models

Three models were chosen to demonstrate the learning abilities of a CNN for FLD. Only a small amount of pre-processing was carried out on the related data as considering the relatively small size of the training set, a CNN should be able to isolate the necessary features. Points' locations were clipped to be between 0 and 250 as some annotations were in negative regions and both the images and points arrays were normalized. This is done for accelerating optimization [7]; Inputting features on varying scales can make convergence slower as the optimizer function struggles

to find optimal points. Normalizing also helps to prevent early saturation if using non-linear activation functions. For this paper images were not made into gray-scale before passing them into the CNN as the input data was not noisy enough to warrant it.

3.1. Initial Model

Simplicity was key for the initial build as a faster model helps in getting used to the data. Built using TensorFlow's Keras API the model consisted of one convolutional layer (C1) with a recitifed linear (ReLU) activation function, followed by a maxpooling layer as seen in 3. C1 has kernel size 3x3 and MP1 has a pool-size of

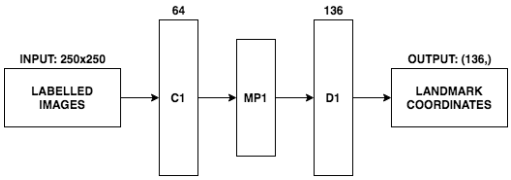

Figure 3. Model One Layers Breakdown

2x2. 136 landmark values, reshaped to (68,2), are the direct output (after being flattened) of our final fully connected layer, D1. Loss was calculated using MSE to highlight failure if a landmark was too far from its ground truth and ADAM was the model's optimizer due to its efficiency and memory requirements [9]. ReLU was the activation function choice for all convolutional layers throughout this paper due to its non-saturated attributes, giving it the ability to increase convergence speed and solve the vanishing gradient problem [16]. Other similar functions do exist however it was not within the scope of this paper to explore their effects on performance. Trained for three epochs in under 3 minutes as expected this models performance was average at best 4. This average face

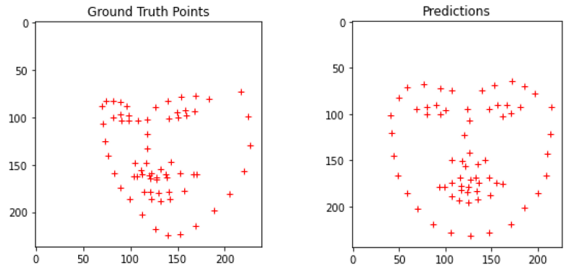

Figure 4. Example of Predicted Face Landmarks Using Model One

was consistent across many image, landmark predicted pairs and is a sign that more convolutional layers are required for the model to learn the characteristics of the training data. Euclidean Distance was selected at the metric to quantify error across landmarks and sets of landmarks due to its simple calculation of an absolute difference between points. Using this metric Model One yielded an average of 12 across all points in all images 5. Eyes performing much better than average and the facial bounds/chin area performing the worst - a contrast between the most densely and most loosely packed landmarks of the face.

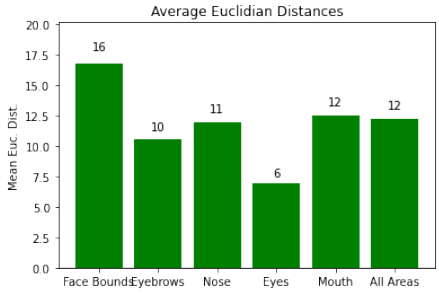

Figure 5. Model One: Euclidean Distances across Facial Areas

3.2. Model Two

Learning from our previous model more layers were implemented so the second network consists of a twice repeated structure of two convolutional layers and a max-pooling, with a final dense layer again 6. Kernel sizes and pool sizes match that of Model One but in our second set of convolutional layers (C3,C4) the filter size is increased from 64 to 98. This initially gave worse

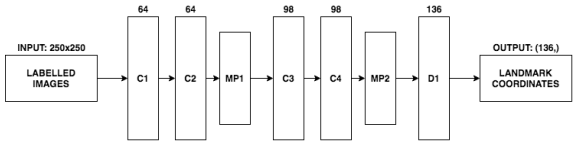

Figure 6. Model Two Layers Breakdown

results however it may have been due to the data augmentation that took place prior. In order to further expand the models learning capabilities three types of data augmentation were implemented: A mirror which flipped image and points in the x direction, rotation by either 5 or 10 degrees clockwise/anticlockwise and a pixel intensity augmenter 7.

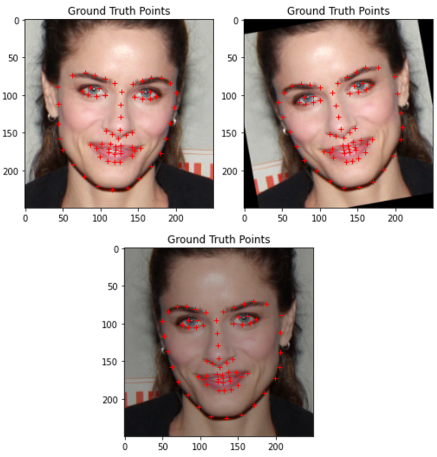

Figure 7. Image Augmentation: Mirror, Rotation, Pixel Intensity

At first 500 of these augmented images were created and at that amount the worst results were observed, with an average Eu-

clidean Distance of 16.0.

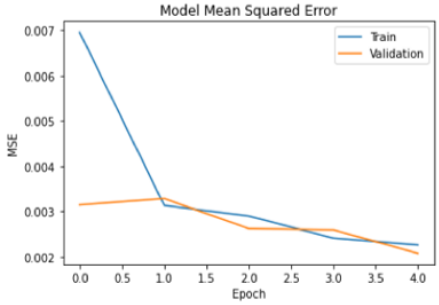

Figure 8. Model Two: MSE - Training/Validation Data (Augmented) split of 80/20

The downward trend in 8 is a positive result however the scale of the y-axis would suggest that the model is under-fitting and the network has not yet learnt the relevant patterns. One solution to this would be to increase the amount of epochs but that can become very time consuming. And so augmentation was removed which yielded a more positive result in average Euclidean Distance 9 with an average of 8.90 and also a similar breakdown of performance across different areas of the face as seen in model one.

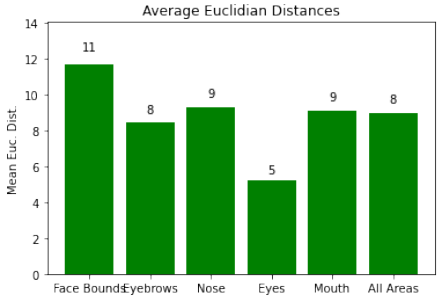

Figure 9. Model Two: Euclidean Distances across Facial Areas

Comparing the models mse without the additional augmented data shows the model is now a good fit - validation error is low and training error only slightly higher 10. Whilst data augmentation is important to overcome the restriction of training data [15] our augmentations here lacked certain realistic variations and so the domain of this synthesised data possibly had a different distribution to our training domain [15].

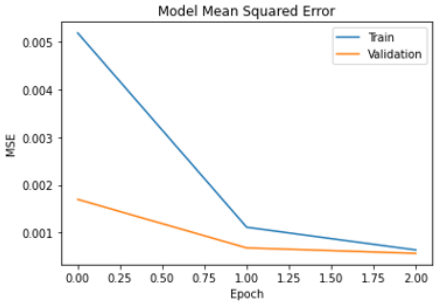

Figure 10. Model Two: MSE - Training/Validation Data (NOT Augmented) split of 80/20

3.3. Final Model

Taking what was learnt from the previous networks, the third and final model consists of two repeated sets of 3 convolution layers and 1 maxpooling layer with a final dense layer 11. This model was trained only on the original data supplied and as over-fitting never arose as an issue, the final model like the previous two, is without a dropout layer. The convolutional layers do all have the "valid" parameter passed to their padding argument and the final activation function attached the last fully connected layer is the Sigmoid function. With a batch size of 32 this model did take the

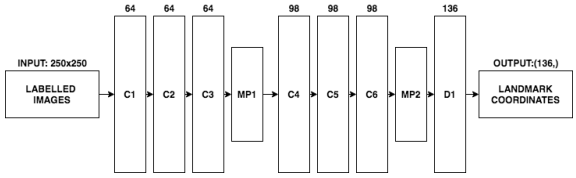

Figure 11. Model Three Layers Breakdown

longest to train however also returned the best results achieving an average Euclidean Distance of 6.58 12. By increasing the convolution layers, we have increased the detail of features that the network is able to pick up.

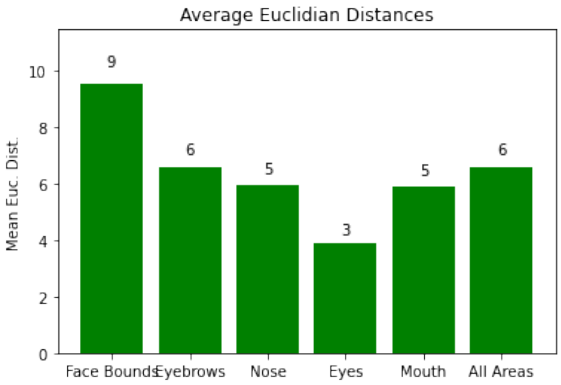

Figure 12. Model Three: Euclidean Distances across Facial Areas

This is a near double improvement in performance from our initial model (Average: 12) and whilst we have traded the CNNs speed, it was a justified swap as now predictions can map to the true landmarks, rather than the average face - as seen in 13 which showcases some example images. They are fairly representative of performance in our bar plot, with eye landmarks being accurately marked and areas around the face being the hardest to predict.

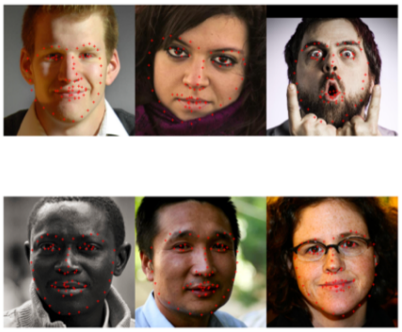

Figure 13. Example Images with Model Three Predicted Points

4. Face Segmentation and a Fun Filter

Segmenting areas of the face can be achieved with a variety of methods but considering this papers focus was predicting landmarks, we chose to use a thresholding-based method. By drawing a polygon around certain predicted landmark's we can augment that areas pixels' values and then create various masks by overwriting the rest of the image's values. 14.

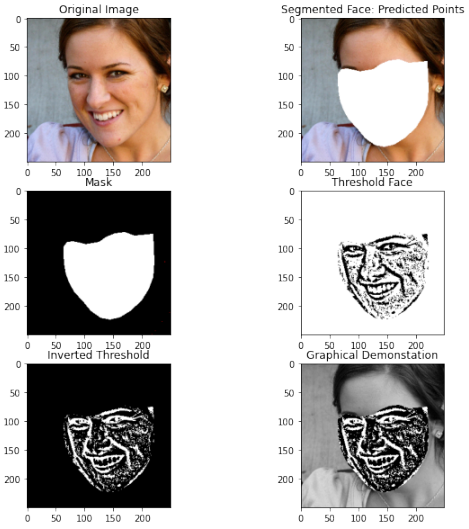

Figure 14. Face Segmentation and Graphical Effect using Predicted Landmarks

Details of the process to create the final image (inverted binary mark added on an averaged image) can be seen in 15.

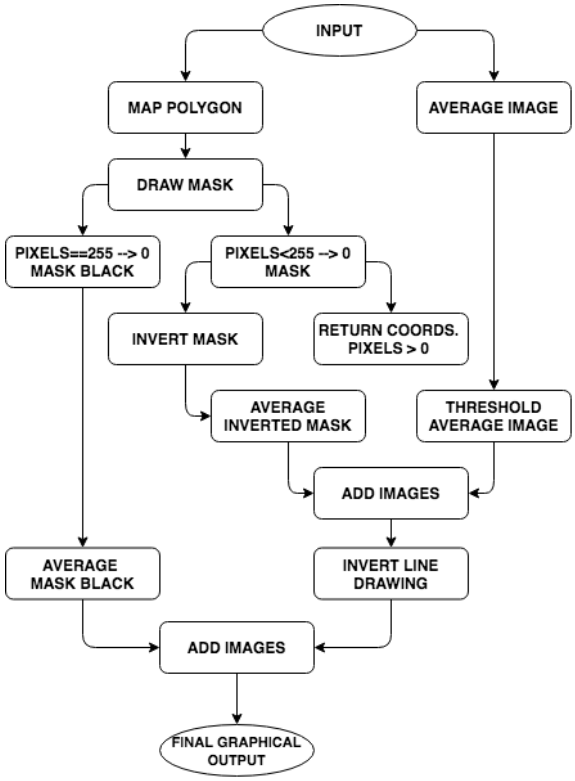

Figure 15. Flowchart detailing steps to create Graphical Demonstration

5. Improvements

Improvement in two main areas was identified. The first being the counter-productive data augmentation. Introducing transfer learning techniques or domain adaptation might have helped synthetic data be of use to our model [11, 6]. However, increasing the size of the images set would have made the already timely training process even longer, making the CNN impractical to use. To mitigate this, the second area of improvement, would be to experiment with dropout layers, which could help keep the models training time low and prevent the generalization of the synthetic data.

6. Conclusion

In this paper we have shown a few models and their qualitative performances and given evaluation metrics to demonstrate performance quantitatively. Our final model gave acceptable results and showed a clear improvement from the first, with predictions in most areas of the face having a Euclidean Distance of six or below. The CNN approach does appear to be the most accessible and with some theory and experimentation, can return good results.

References

[1] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. *International Journal of Computer Vision*, 107(2):177–190, 2014. 1

[2] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence*, 23(6):681–685, 2001. 1

[3] V. Dumoulin and F. Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016. 1

[4] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org. 1

[5] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012. 1

[6] S. Hong, W. Im, J. Ryu, and H. S. Yang. Sspp-dan: Deep domain adaptation network for face recognition with single sample per person. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 825–829. IEEE, 2017. 4

[7] J. Jin, M. Li, and L. Jin. Data normalization to accelerate training for linear neural net to predict tropical cyclone tracks. *Mathematical Problems in Engineering*, 2015:1–8, 07 2015. 1

[8] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. 06 2014. 1

[9] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, May 2017. 1

[11] B. Liu, X. Wang, M. Dixit, R. Kwitt, and N. Vasconcelos. Feature space transfer for data augmentation. In *Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9090–9098. IEEE Computer Society, 12 2018. 4

[12] D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, page 2879–2886, USA, 2012. IEEE Computer Society. 1

[13] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1685–1692, 2014. 1

[14] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, page 3476–3483, USA, 2013. IEEE Computer Society. 1

[15] X. Wang, K. Wang, and S. Lian. A survey on face data augmentation. *arXiv preprint arXiv:1904.11685*, 2019. 3

[16] B. Xu, N. Wang, T. Chen, and M. Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015. 2

[17] X. Zhu, X. Liu, Z. Lei, and S. Li. Face alignment in full pose range: A 3d total solution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP:1–1, 11 2017. 1