Multi-task learning of facial landmarks and attributes with Tensorflow

Felix Abrahamsson, Joar Gruneau, Martin Zuber

KTH Royal Institute of Technology

Abstract. This paper focuses on investigating methods of optimizing facial landmark detection through the use of multi-task convolutional neural networks. The idea behind the multi-task approach is to learn facial landmarks jointly with related, auxiliary tasks such as head pose estimation. By doing so one can take advantage of the similarity between the tasks to increase the accuracy of facial landmark detection. To avoid the problem of over fitting due to different learning rates of the tasks, early stopping of the auxiliary tasks was implemented. The report compares accuracies for multi- and single-task facial landmark detection. Based on the results, the multi-task approach was not found to improve the performance on facial landmark detection compared to the single-task approach. This contradicts earlier findings, and possible reasons for these results are discussed.

1 Introduction

At its core, this project tackles a multi-task convolutional network running on images of human faces. The learning is focused on two tasks:

- Determining the location of several facial landmarks: eyes, nose and corners of the mouth. There are therefore five locations in total.
- Recognizing certain facial attributes: gender, smile, glasses, which are all binary classification problems, as well as head pose which comprises five positions.

Performance *per se* on the dataset is not the focus of this project. Of course the network is built to be efficient at handling the tasks at hand but, in the end, focus will be on the multi-tasking aspect of the network and how it can affect performance. Our hope at the start was to be able to increase performance on the landmark recognition and the attribute recognition individual tasks by learning them simultaneously with the same network. Even though a multi-task learning approach has been shown to increase performance in *a priori* unrelated tasks (see Romera-Paredes et al., 2012 [10]), our hope here was that the obvious correlation between the attributes and the landmarks could lead to a successful implementation of a multi-task network.

This report aims to give an account of the way the design and implementation of the network was handled, as well as of the issues encountered on the way. We

will go into more detail about the design choices that were made for the network and different issues that arose from transforming a single-task network into a multi-task one (see sections 2 and 3). For the actual implementation of the network Python and the Tensorflow library [1] were used, which had its own set of challenges (see section 3).

2 Background

2.1 Multi-task learning

Multi-task learning has been used in many computer vision tasks [5] [6] [7], but also in speech recognition [8]. In 1997, Rich Caruna gave the following description of multi-task learning and its benefits [9]:
"*Multi-task Learning is an approach to inductive transfer that improves generalization by using the domain information contained in the training signals of related tasks as an inductive bias. It does this by learning tasks in parallel while using a shared representation; what is learned for each task can help other tasks be learned better.*"
Using this shared representation for different tasks also helps reduce the size and complexity of the model, allowing one to make due even with limited computing resources.

Often a *main* task is defined, on which performance is evaluated. The other tasks are referred to as *auxiliary* tasks, and serve to raise performance on the main task [9]. In this project, the main task is considered to be facial landmark detection, while the attribute detection tasks comprise the auxiliary tasks.

2.2 Previous approaches to landmark detection

Many different approaches to facial landmark detection have been proposed over the past few years. Sun *et al.* proposed an approach based on cascaded convolutional networks in 2013, at the time outperforming state-of-the art results [3]. They built a deep network, consisting of three levels of convolutional networks where the outputs were fused at each level.

In 2014, Zhang *et al.* proposed an approach based on multi-task learning which outperformed the cascaded convolutional network approach by Sun *et al.* [2]. Their network was trained to predict both facial landmarks and a combination of other facial attributes such as gender, wearing/not wearing glasses, smiling/not smiling and different head poses at the same time. In their evaluation, facial landmark detection was considered as the main task. They argued that since the tasks were correlated, the network would benefit from learning the task simultaneously. For instance, a 60° head pose results in a smaller interocular distance than that of a frontal image, a smile results in different positions of the mouth corners. They were able to show that their network implicitly learned relationships between two related tasks, and that performance on facial landmark detection was improved by employing a multi-task approach.

2.3 Defining loss functions

To train the network on attribute detection the cross-entropy loss for a mini-batch was used [14]:

$$L_A(\mathbf{X}, \mathbf{Y}, \Theta) = \frac{\sum_{i=1}^{N} l_{cross}(\mathbf{x}_i, \mathbf{y}_i, \Theta)}{N} = \frac{\sum_{i=1}^{N} -\log(\mathbf{y}_i^T * p(\mathbf{x}_i, \Theta))}{N}, \quad (1)$$

where the number N denotes the size of the mini-batch. Here $p(\mathbf{x}_i, \Theta)$ is the softmax function, $\mathbf{y}$ is the one hot encoding of the attribute and Θ are the network weights. For the facial landmarks a least square loss was used [2]:

$$L_{FL}(\mathbf{X}, \mathbf{Y}, \Theta) = \frac{1}{2N} \sum_{i=1}^{N} \sum_{j=1}^{K} ||\mathbf{y}_i^j - f(\mathbf{x}_i^j, \Theta)||^2. \quad (2)$$

Here j denotes one of the K landmarks, y_i^j is the ground truth landmark position of landmark j for data point i, $f(\mathbf{x}_i^j, \Theta)$ is the predicted position of the network.

For the multi-task network the losses can be combined from the single task networks into a joint loss, as defined in [2]:

$$L_{Multi}(\mathbf{X}, \mathbf{Y}, \Theta) = L_L(\mathbf{X}, \mathbf{Y}, \Theta) + \sum_a \lambda^a * L_A^a(\mathbf{X}, \mathbf{Y}, \Theta). \quad (3)$$

Here L_{FL} is the loss for facial landmarks and L_A^a is the loss for attribute a. The coefficient λ^a is the importance coefficient for attribute a, which controls the relative contribution of the loss of task a to the total loss.

2.4 Early stopping

The approach by Zhang *et. al* focuses on obtaining the best possible accuracy for facial landmarks by jointly learning related auxiliary tasks such as wearing glasses, smiling, gender and head pose. Since the learning converges differently for the main and the auxiliary tasks it is necessary to implement some form of early stopping of the auxiliary tasks. If no such technique is applied, classifiers for the auxiliary tasks would overfit and potentially influence performance on the main task negatively. A method for early stopping was proposed by Zhang *et. al*, where they would halt the auxiliary task if a certain measure exceeded a threshold ϵ:

$$\frac{k * med_{j=t-k}^t(L_{tr}^t(j))}{\sum_{j=t-k}^t L_{tr}^t(j) - k * med_{j=t-k}^t(L_{tr}^t(j))} * \frac{L_{val}^t - min_{j=t-k}^t(L_{val}^t(j))}{\lambda^a * min_{j=t-k}^t(L_{val}^t(j))} > \epsilon, \quad (4)$$

where L_{tr}^t and L_{val}^t is the training and validation loss for iteration t. The measurement $med_{j=t-k}^t(L_{tr}^t(j))$ is the median of the training loss over the last k iterations. λ^a is the importance coefficient for learning task a as mentioned above. The value ϵ is the halting threshold for the auxiliary task.

2.5 Network architecture

The architecture/design of a CNN is a critical matter, and depends on the task at hand. Coates *et. al* analyzed different architectures for CNNs to be used for image recognition and feature learning [4]. They concluded that computational power should be focused on reducing stride size rather than making the network learn more features, should the computational power be an issue. In their experiments, a receptive field size (filter size) of 6 appeared to give the best performance.

Zhang *et. al* used a 4-layer convolutional network in their design, each convolutional layer followed by a max-pooling layer. Finally, a fully-connected layer was applied for feature extraction, shared by 5 separate layers each handling separate tasks.

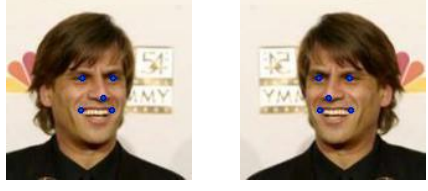
3 Approach

3.1 Dataset acquisition and pre-processing

To train our models, the publicly available MTFL dataset was used [11]. The dataset consists of RGB images of faces annotated with the facial landmarks and attributes mention in section 1, and is split into three separate sets: LFW, NET and AFLW. A few of the images were rectangular and of non portrait type and some were only black and white. These images were excluded. All images were also resized to 150x150.

The images used for training were taken from the LFW and the NET sets (7650 images) . For validation, 1000 images were taken from the AFLW set, while the rest (1994 images) were used for testing. It should be noted that the AFLW set is considered more difficult to predict, as 39% of the faces are non-frontal and many faces are partially occluded [2].

Some data augmentation was applied to the training set as well, where images were translated horizontally. An example of this can be seen in figure 1, annotated with the translated ground-truth facial landmarks.



(a) The raw image.

(b) The augmented image.

Fig. 1: Visualization of the data augmentation, faces annotated with the ground-truth facial landmarks.

3.2 Network architecture and design choices

In total, three networks were implemented using Tensorflow. The first one serves to predict facial landmarks as a standalone task, the second one to predict facial attributes one at a time, and the final network to predict facial landmarks and one or more attributes at the same time. All three networks share a common architecture structure. First, two convolutional layers are applied. The first layer consists of 16 filters of size 5x5x3 (RGB images $\Rightarrow$ 3 input channels), each employing stride 1 with padding to retain image size. The first layer thus outputs 16 response maps of size 150x150. Before the second convolutional layer, a max-pooling layer was applied, halving the size of the response maps to 75x75. The second convolutional layer consists of 32 filters, each of size 5x5x16 with stride 1 and applying padding to retain size. After the second convolutional layer, another max-pooling layer was applied so that the final output of the convolutions were 32 response maps of size 38x38. Finally, a fully connected layer followed by a dropout layer was applied for feature extraction, outputting a feature vector of size 1024. This feature vector was shared among all tasks, and served as the input to the classification and regression layers which would output either facial landmark coordinates or facial attributes.

Different architectures were experimented with to some extent, such as increasing the amount of filters, the size of the output feature vector, or the amount of convolutional layers. However, due to the large amount of degrees of freedom in the design and the high computational cost of experiments, it was not possible to perform an exhaustive search for a good network architecture. In the end, the design above was selected as a compromise between computational cost and performance. Experiments were run on a Nvidia GTX 950M GPU card, and increasing the network size would result in a very constrained mini-batch size for the SGD because of the memory limitations of the GPU. However, as performance was not of top priority for this project, the architecture was deemed to perform sufficiently well.

3.3 Weight initialization and training

For training, as a naive first approach weights were initialized from a normal distribution and the bias terms were initialized with a slight positive constant bias. However, this resulted in highly unstable initial loss values for the attribute classifiers. To solve this problem, Xavier initialization was used for the weights [12], which stabilized the loss values in a satisfactory manner. To update the weights, stochastic gradient descent was applied using Tensorflow's built-in Adam optimizer to achieve a good adaptive learning rate [13]. The loss functions used by the optimizer were as described in section 2.3.

Our implementation of early stopping for the auxiliary tasks followed equation (4), with $\epsilon = 15 \cdot \lambda^a$ for task a. The condition was tried for each task once per epoch.

3.4 Experiments

Initially, a dropout layer was implemented in the network to add regularization. However, it turned out that performance always suffered from added dropout, so in the end it was completely omitted for all experiments. An L2 regularization term on the weights were also added, but it again resulted in a performance decline, so it was removed.

Since there is no definitive measurement of accuracy on facial landmarks, the definition given by Zhang *et. al* [2] was used. The inter-ocular distance (the distance between the eyes) was measured, and normalized the distance between the ground truth positions of the landmarks and the positions predicted by the network by this distance. If this relative error was above 10%, the predicted position was considered a failure.

A decision was made not to train the multi-task network on smile and gender recognition, as the single-task learner did not perform very well on those tasks. So to summarize, the multi-task learner was trained to predict facial landmarks, head pose and glasses.

The importance coefficients were chosen as $\lambda^{pose} = 6.0$ and $\lambda^{glasses} = 3.0$. A higher value for the importance coefficient for pose estimation were chosen because it was believed that the pose is more closely related to learning landmarks. The reasoning is that head pose affects the positions of all facial landmarks relative to one another simultaneously, while glasses only affect the look of the eyes, and does not affect their positions significantly. Note also that the coefficient λ^a controls the contribution of task a to the join loss function, as defined by equation (3). As the values of the loss for the landmarks is on the order of $30 - 60$, while the loss for the attributes is on the order of $0.2 - 1.5$ (slightly larger for pose estimation than for glasses) it is necessary to have $\lambda^a >> 1$. However, it was noted that larger values of the λ^a lead to a decline in performance for the facial landmarks.

The following experiments were run:

- Single-task network on head pose
- Single-task network on glasses
- Single-task network on facial landmarks
- Multi-task network without early stopping
- Multi task network with early stopping
- Multi-task network without early stopping and with augmented data

The networks were trained for 30 epochs and ran every experiment three times (with different initializations), then averaged the accuracy to obtain more reliable results for evaluation.

4 Results and Conclusions

4.1 Accuracy measurements and evaluation

Figure 2 shows the mean accuracy on facial landmark detection for the different networks, measured on the test set. "S-T" is an abbreviation for single task, "M-T" for multi task, "ES" for early stopping and "aug" for "augmented data".

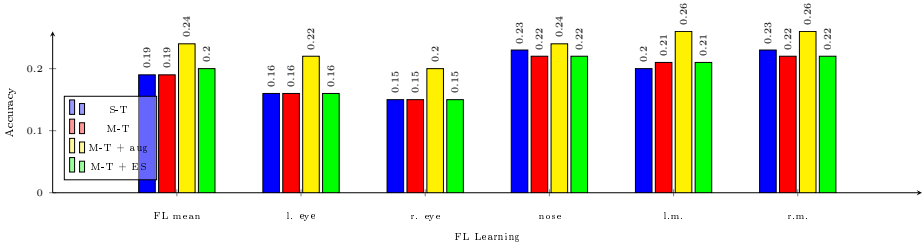


Fig. 2: Mean accuracy for 30 epochs for the different facial landmarks through different learning processes

From figure 2 we can see that the single-task and multi task networks achieve very similar accuracy on facial landmarks. This is unfortunate but not devastating: such a network would take fine tuning to be adapted to the specific set of data on which it runs.

During training the learning of the *glasses* attribute was stopped after 11.3 epochs on average and *pose* was stopped after 15.5 epochs on average. This indicates that classifying head pose is a more challenging task than classifying glasses. However, it is difficult to say whether this is because the poses were labeled in five different ways while glasses was a binary classification problem, or simply because pose estimation in itself is intrinsically more difficult than detecting glasses.

Figure 3 shows the accuracy on the different facial landmarks, as well as on the two attributes for the multi-task network with and without early stopping applied. For this particular trial, pose estimation was stopped at the 2:nd epoch, and as a result the accuracy on pose estimation is very poor (similar to a random guesser). Clearly, the early stopping was applied too early and the network failed to learn pose estimation. This indicates instability in the implementation of the early stopping criteria.

Based on the results from figures 2 and 3 it does not appear that early stopping improved the results of the network by any significant amount. It could be that the selected threshold was too low, or because of other instabilities in the network. Figure 4 shows the validation accuracy of the different facial landmark classifying networks over 30 epochs. While the plot shows an overall increase in accuracy for all networks over time, the accuracy fluctuates from epoch to

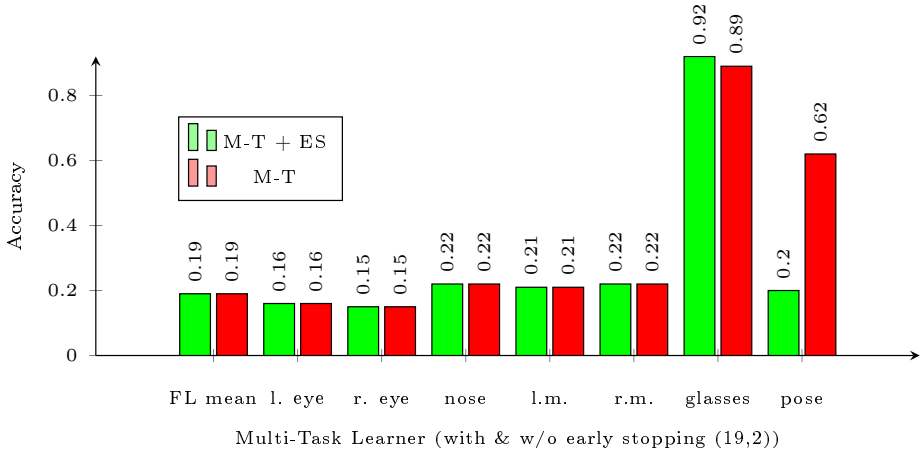


Fig. 3: Early stopping at 2 for the pose and 19 for the glasses

epoch by a significant amount. This could explain why the early stopping criteria appears to be unstable as well, as it relies on the measured validation loss (see equation (4)).

Without a good intuition, finding the right balance for early stopping takes time as it requires training with different settings every time and checking the results ideally on the average for multiple trials for each setting. This is where a good chunk of the work would still have to be done. The results on early stopping only show the potential that testing every possibility (ideally with augmented data) would have for the construction of a more efficient network. However our results here show the reality of that potential and that is where our goal was.

One other notable result is that the multi-task network with augmented data outperforms the other networks by a significant margin in figure 2 as well as in figure 4. This shows that the natural limit of the network may not have been reached as an increase in the dataset size results in a significant increase in the accuracy on the testing set. Figure 5 shows a comparison between the performance of the multi-task network and the multi-task network with augmented data.

As we can see from figure 4 the accuracy has not yet converged but is still increasing. Since early stopping eliminates the risk of over-fitting auxiliary tasks it is possible that we would have obtained better results if we continued the training for a longer time. However since these experiments required great computational resources and took almost a full day to run we did not have time to re-run all of the tests with many more epochs. The single-task network does appear to be converging slightly faster, which could indicate that the multi-task network could possibly benefit more from further training and surpass the per-

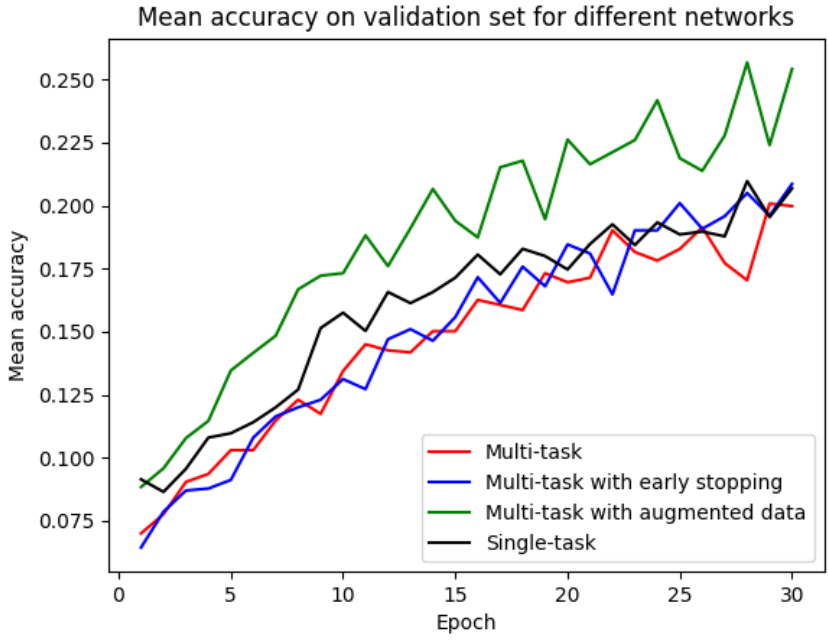


Fig. 4: Mean accuracy of the facial landmarks on the validation set for different networks over 30 epochs.

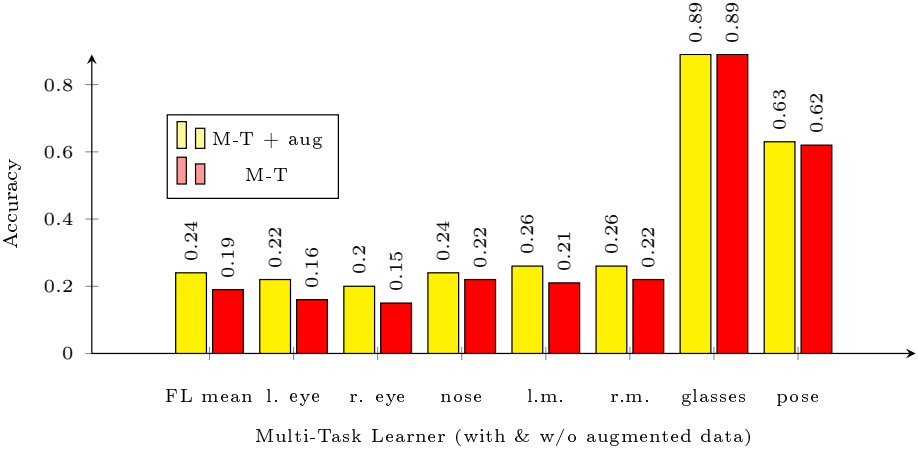


Fig. 5: Mean accuracy with or without the use of augmented data

formance of the single-task network. However, the difference is quite small and could very well be within the standard deviation.

As we can see the accuracies for facial landmark detection are quite poor whichever network architecture we choose. For all networks we performed a sanity check and evaluated the network on the training data. All networks were able to over-fit on the training data and obtain an accuracy of nearly 100%, both on facial landmark detection and facial attributes estimation. This suggests that the poor accuracy is not a result from an incorrect network architecture or a bug in the code. As mentioned in [2] the data set used for testing and validation is more difficult to predict and has a larger amount of pose variation (non frontal images) and severe partial occlusion. This could explain the low accuracies on the test and validation set.

It should still be noted that the multi-task learner did not perform worse than the single-task learner. Considering that the complexity of what the multi-task learner needs to learn is higher it is quite possible that more training is necessary, so perhaps it is not entirely fair to compare the two networks trained for the same amount of epochs.

4.2 Facial landmark annotation by the different networks

As a final method of evaluation, we let the networks annotate images from the test set by marking the facial landmarks in the images with dots. The results can be seen in figure 6. On all images, the multi-task network using augmented data performs the best.

The top row in figure 6 shows an image with poor resolution, on which all networks seem to perform similarly well. The second row shows an image with more shading and some pose variation, on which the single task network seems to perform better than the multi-task network. The image on the third row is a simple, frontal image, and is arguably the easiest image to predict. This is confirmed by the output of the networks as well, as they all appear to be able to predict the facial landmarks with almost perfect precision. The fourth row shows an image with heavy pose variation, on which the only notable difference in performance is that of the network using augmented data. The fifth and final row shows an image with a person wearing glasses, as well as with some pose variation. On this image, one would expect the multi-task networks to perform much better than the single-task network, since the multi-task networks were trained both on detecting glasses and pose variation. The multi-task networks do appear to estimate the mouth corners with better precision, however the estimation of the eyes is not improved.

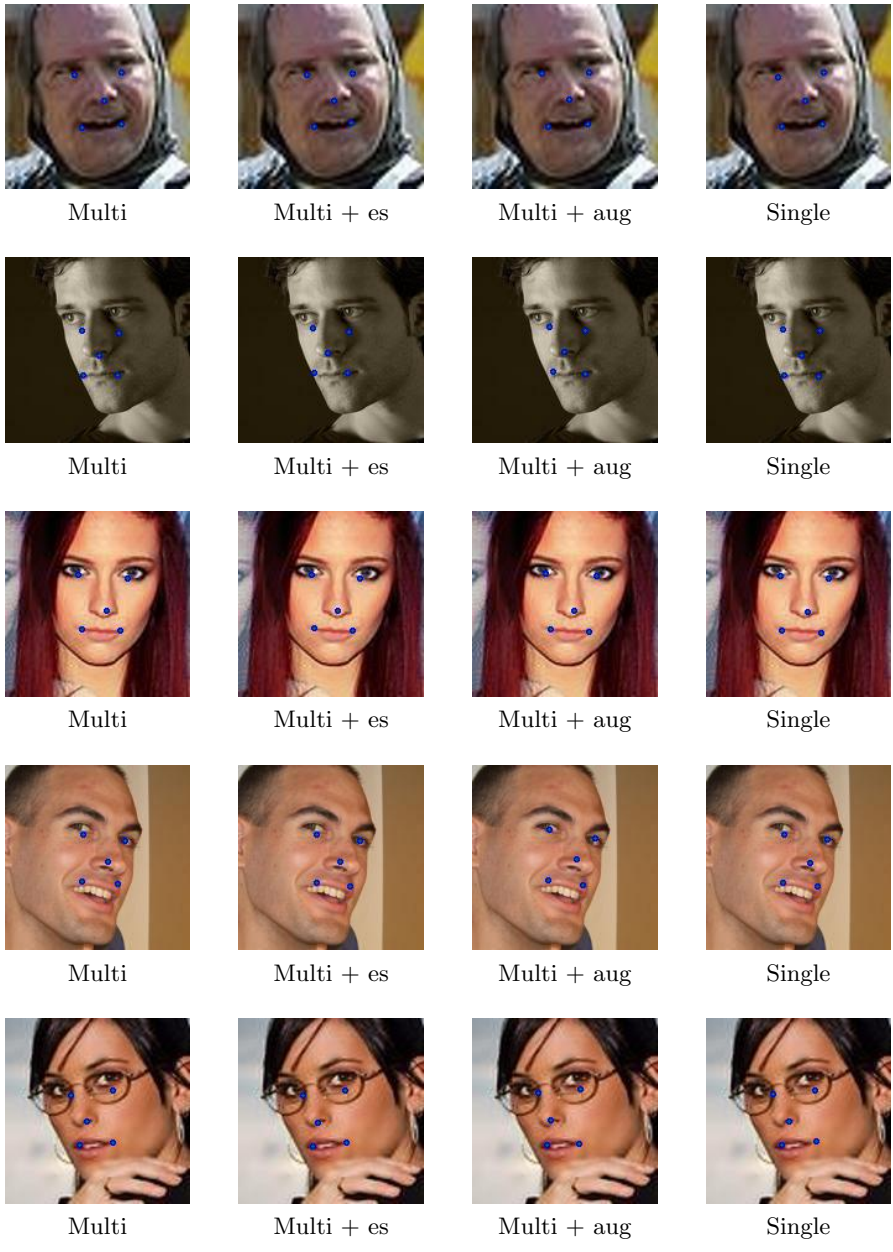


Fig. 6: Facial landmark annotation on the test set by the different networks after training for 30 epochs. Abbreviations: 'es' = early stopping, 'aug' = augmented data included.

4.3 Final conclusions

To summarize, the multi-task network does not show any significant improvement over the single-task network on facial landmark detection. It is possible that training for a longer period of time with much heavier data augmentation, coupled with a more complex network would show greater differences, but due to the limited time frame of this project this was not possible to test.

It should also be noted that the multi-task network did not increase the computational cost of training by any significant amount, due to the large amount of shared features for the different tasks. Therefore, as performance did not decline through the use of the multi-task approach, it can still be a desirable option to the single-task approach if one wants to learn multiple tasks at the same time, not just for the sake of improving the main task.

It is also possible that the importance coefficients λ^a need to be fine-tuned more. The only metric used in this project to determine reasonable values for the λ^a was the order of magnitude of the different loss functions, and while that is a sensible metric it may be necessary to experiment more to find out exactly how to weigh the loss terms. Furthermore, it might be beneficial to implement adaptive λ^a coefficients, though that does of course entail more complexity in the model.

References

1. *Tensorflow*, URL: https://www.tensorflow.org/, accessed May 4th, 2017.
2. Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang, *Facial Landmark Detection by Deep Multi-task Learning*, Dept. of Information Engineering, The Chinese University of Hong Kong, 2014.
3. Yi Sun, Xiaogang Wang, Xiaoou Tang, *Deep Convolutional Network Cascade for Facial Point Detection*, The Chinese University of Hong Kong and the Chinese Academy of Sciences, 2013.
4. Adam Coates, Honglak Lee, Andrew Y. Ng, *An Analysis of Single-Layer Networks in Unsupervised Feature Learning*, Stanford University and University of Michigan, 2011.
5. Xi Yin, Xiaoming Liu, *Multi-Task Convolutional Neural Network for Pose-Invariant Face Recognition*, IEEE, 2017.
6. Xiao-Tong Yuan, Shuicheng Yan, *Visual Classification with Multi-Task Joint Sparse Representation*, National University of Singapore, 2012.
7. Tianzhu Zhang, Bernard Ghanem, Si Liu, Narendra Ahuja, *Robust Visual Tracking via Structured Multi-Task Sparse Learning*, IJCV 101(2), 367383, 2013.
8. Abhinav Thanda, Shankar M Venkatesan, *Multi-task Learning Of Deep Neural Networks For Audio Visual Automatic Speech Recognition*, Samsung R&D Institute India, 2017.
9. Rich Caruna, *Multitask Learning*, Carnegie Mellon University, 1997.
10. Exploiting Unrelated Tasks in Multi-Task Learning Bernardino Romera-Paredes, Andreas Argyriou, Nadia Bianchi-Berthouze, Massimiliano Pontil, *Exploiting Unrelated Tasks in Multi-Task Learning*, 2012.
11. *The MTFL dataset*, URL: http://mmlab.ie.cuhk.edu.hk/projects/TCDCN.html, accessed May 4th, 2017.

12. Xavier Glorot, Yoshua Bengio. *Understanding the difficulty of training deep feedforward neural networks*, Aistats. vol. 9, 2010.
13. Diederik P. Kingma, Jimmy Ba, *Adam: A Method for Stochastic Optimization*, 3rd International Conference for Learning Representations, San Diego, 2015
14. Ian Goodfellow and Yoshua Bengio and Aaron Courville. *Deep Learning* http://www.deeplearningbook.org