Deep Alignment Network: A convolutional neural network for robust face alignment

Marek Kowalski, Jacek Naruniec, and Tomasz Trzcinski

Warsaw University of Technology

m.kowalski@ire.pw.edu.pl, j.naruniec@ire.pw.edu.pl, t.trzcinski@ii.pw.edu.pl

Abstract

In this paper, we propose Deep Alignment Network (DAN), a robust face alignment method based on a deep neural network architecture. DAN consists of multiple stages, where each stage improves the locations of the facial landmarks estimated by the previous stage. Our method uses entire face images at all stages, contrary to the recently proposed face alignment methods that rely on local patches. This is possible thanks to the use of landmark heatmaps which provide visual information about landmark locations estimated at the previous stages of the algorithm. The use of entire face images rather than patches allows DAN to handle face images with large variation in head pose and difficult initializations. An extensive evaluation on two publicly available datasets shows that DAN reduces the state-of-the-art failure rate by up to 70%. Our method has also been submitted for evaluation as part of the Menpo challenge.

1. Introduction

The goal of face alignment is to localize a set of predefined facial landmarks (eye corners, mouth corners etc.) in an image of a face. Face alignment is an important component of many computer vision applications, such as face verification [28], facial emotion recognition [25], human-computer interaction [6] and facial motion capture [12].

Most of the face alignment methods introduced in the recent years are based on shape indexed features [33, 22, 4, 36, 30]. In these approaches image features, such as SIFT [33, 36] or learned features [22, 30], are extracted from image patches extracted around each of the landmarks. The features are then used to iteratively refine the estimates of landmark locations. While those approaches can be successfully applied to face alignment in many photos, their performance on the most challenging datasets [23] leaves room for improvement [4, 33, 36, 30]. We believe that this is due to the fact that for the most difficult images the features extracted at disjoint patches do not provide enough information and can lead the method into a local minimum.

In this work, we address the above shortcoming by proposing a novel face alignment method which we dub Deep Alignment Network (DAN). It is based on a multi-stage neural network where each stage refines the landmark positions estimated at the previous stage, iteratively improving the landmark locations. The input to each stage of our algorithm (except the first stage) are a face image normalized to a canonical pose and an image learned from the dense layer of the previous stage. To make use of the entire face image during the process of face alignment, we additionally input at each stage a *landmark heatmap*, which is a key element of our system.

A landmark heatmap is an image with high intensity values around landmark locations where intensity decreases with the distance from the nearest landmark. The convolutional neural network can use the heatmaps to infer the current estimates of landmark locations in the image and thus refine them. An example of a landmark heatmap can be seen in Figure 1 which shows an outline of our method. By using landmark heatmaps, our DAN algorithm is able to reduce the failure rate on the 300W public test set by a large margin of 72% with respect to the state of the art.

To summarize, the three main contributions of this work are the following:

1. We introduce landmark heatmaps which transfer the information about current landmark location estimates between the stages of our method. This improvement allows our method to make use of the entire image of a face, instead of local patches, and avoid falling into local minima.

2. The resulting robust face alignment method we propose in this paper reduces the failure rate by 60% on

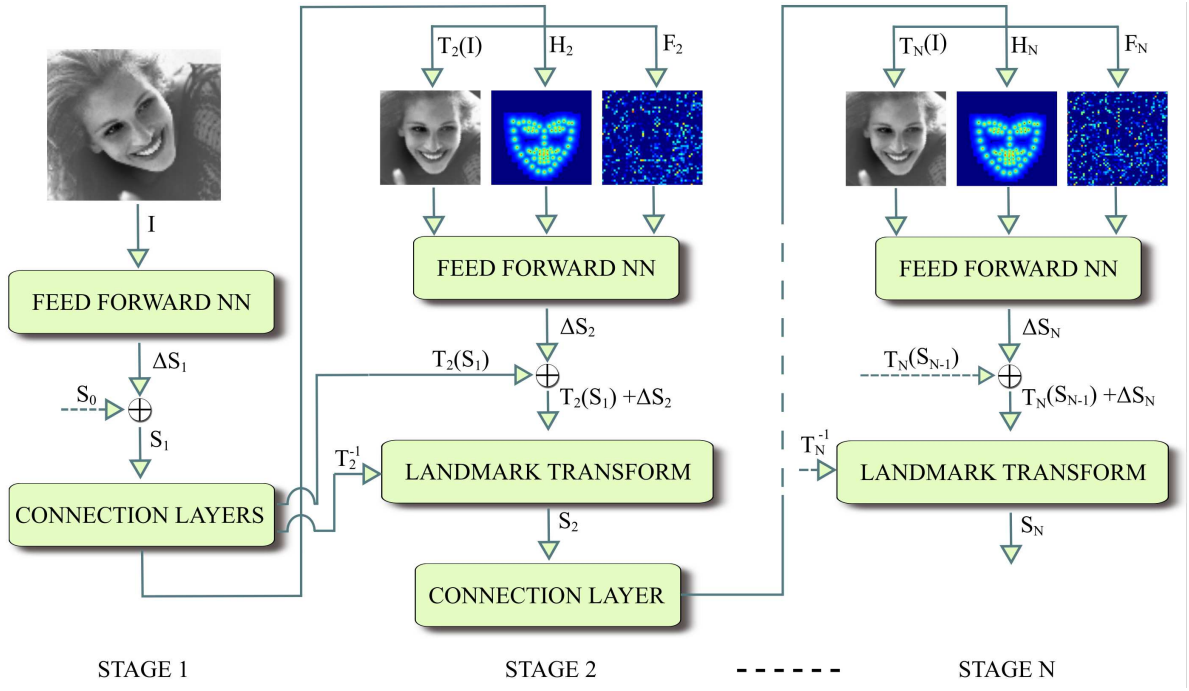

Figure 1. A diagram showing an outline of the proposed method. Each stage of the neural network refines the landmark location estimates produced by the previous stage, starting with an initial estimate S_0. The connection layers form a link between the consecutive stages of the network by producing the landmark heatmaps H_t, feature images F_t and a transform T_t which is used to warp the input image to a canonical pose. By introducing landmark heatmaps and feature images we can transmit crucial information, including the landmark location estimates, between the stages of our method.

the 300W private test set [23] and 72% on the 300-W public test set [23] compared to the state of the art.

3. Finally, we publish both the source code of our implementation of the proposed method and the models used in the experiments.

The remainder of the paper is organized in the following manner. In section 2 we give an overview of the related work. In section 3 we provide a detailed description of the proposed method. Finally, in section 4 we perform an evaluation of DAN and compare it to the state of the art.

2. Related work

Face alignment has a long history, starting with the early Active Appearance Models [5, 21], moving to Constrained Local Models [7, 1] and recently shifting to methods based on Cascaded Shape Regression (CSR) [33, 4, 22, 19, 17, 31, 14] and deep learning [35, 11, 30, 32, 3].

In CSR based methods, the face alignment begins with an initial estimate of the landmark locations which is then refined in an iterative manner. The initial shape S_0 is typically an average face shape placed in the bounding box returned by the face detector[4, 33, 22, 30]. Each CSR iteration is characterized by the following equation:

$$S_{t+1} = S_t + r_t(\phi(I, S_t)), \quad (1)$$

where S_t is the estimate of landmark locations at iteration t, r_t is a regression function which returns the update to S_t given a feature ϕ extracted from image I at the landmark locations.

The main differences between the variety of CSR based methods introduced in the literature lie in the choice of the feature extraction method ϕ and the regression method r_t. For instance, Supervised Descent Method (SDM) [33] uses SIFT [20] features and a simple linear regressor. LBF [22] takes advantage of sparse features generated from binary trees and intensity differences of individual pixels. LBF uses Support Vector Regression [10] for regression which, combined with the sparse features, leads to a very efficient method running at up to 3000 fps.

Coarse to Fine Shape Searching (CFSS) [36], similarly to SDM, uses SIFT features extracted at landmark locations. However the regression step of CSR is replaced with a search over the space of possible face shapes which goes from coarse to fine over several iterations. This reduces the probability of falling into a local minimum and thus improves convergence.

MIX [31] also uses SIFT for feature extraction, while regression is performed using a mixture of experts, where

each expert is specialized in a certain part of the space of face shapes. Moreover MIX, warps the input image before each iteration so that the current estimate of the face shape matches a predefined canonical face shape.

Mnemonic Descent Method (MDM) [30] fuses the feature extraction and regression steps of CSR into a single Recurrent Neural Network that is trained end-to-end. MDM also introduces memory into the process which allows information to be passed between CSR iterations.

While all of the above mentioned methods perform face alignment based only on local patches, there are some methods [35, 11] that estimate initial landmark positions using the entire face image and use local patches for refinement. In contrast, DAN localizes the landmarks based on the entire face image at all of its stages.

The use of heatmaps for face alignment related tasks precedes the proposed method. One method that uses heatmaps is [3], where a neural network outputs predictions in the form of a heatmap. In contrast, the proposed method uses heatmaps solely as a means for transferring information between stages.

The development of novel methods contributes greatly in advancing face alignment. However it cannot be overlooked that the publication of several large scale datasets of annotated face images [23, 24] also had a crucial role in both improving the state of the art and the comparability of face alignment methods.

3. Deep Alignment Network

In this section, we describe our method, which we call the Deep Alignment Network (DAN). DAN is inspired by the Cascade Shape Regression (CSR) framework, just like CSR our method starts with an initial estimate of the face shape S_0 which is refined over several iterations. However, in DAN we substitute each CSR iteration with a single stage of a deep neural network which performs both feature extraction and regression. The major difference between DAN and approaches based on CSR is that DAN extracts features from the entire face image rather than the patches around landmark locations. This is achieved by introducing additional input to each stage, namely a landmark heatmap which indicates the current estimates of the landmark positions within the global face image and transmits this information between the stages of our algorithm. An outline of the proposed method is shown in Figure 1.

Therefore, each stage of DAN takes three inputs: the input image I which has been warped so that the current landmark estimates are aligned with the canonical shape S_0, a landmark heatmap H_t and a feature image F_t which is generated from a dense layer connected to the penultimate layer of the previous stage $t-1$. The first stage only takes the input image as the initial landmarks are always assumed to be the average face shape S_0 located in the middle of the image.

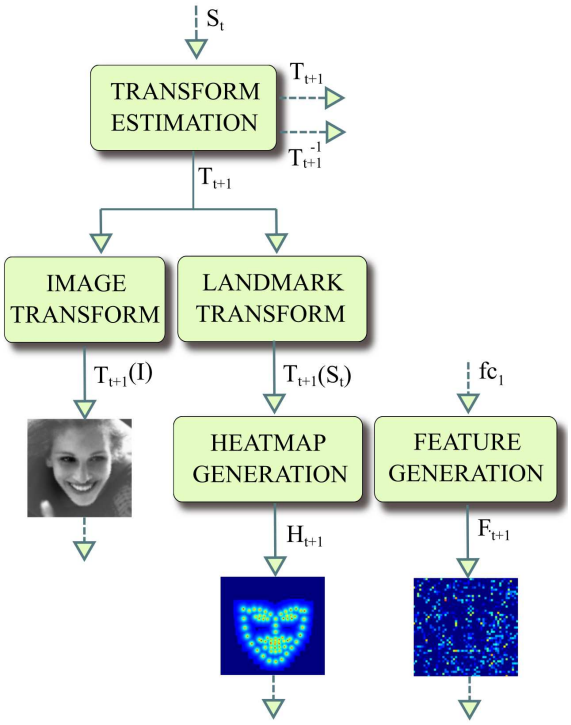

Figure 2. A diagram showing an outline of the connection layers. The landmark locations estimated by the current stage S_t are first used to estimate the normalizing transform T_{t+1} and its inverse T_{t+1}^{-1}. T_{t+1} is subsequently used to transform the input image I and S_t. The transformed shape $T_{t+1}(S_t)$ is then used to generate the landmark heatmap H_{t+1}. The feature image F_{t+1} is generated using the fc1 dense layer of the current stage t.

A single stage of DAN consists of a feed-forward neural network which performs landmark location estimation and *connection layers* that generate the input for the next stage. The details of the feed-forward network are described in subsection 3.1. The connection layers consist of the Transform Estimation layer, the Image Transform layer, Landmark Transform layer, Heatmap Generation layer and Feature Generation layer. The structure of the connection layers is shown in Figure 2.

The transform estimation layer generates the transform T_{t+1}, where t is the number of the stage. The transformation is used to warp the input image I and the current landmark estimates S_t so that S_t is close to the canonical shape S_0. The transformed landmarks $T_{t+1}(S_t)$ are passed to the heatmap generation layer. The inverse transform T_{t+1}^{-1} is used to map the output landmarks of the consecutive stage back into the original coordinate system.

The details of the Transform Estimation, Image Transform and Landmark Transforms layer are described in subsection 3.2. The Heatmap Generation and Feature Image layers are described in sections 3.3, 3.4. Section 3.5 details

Table 1. Structure of the feed-forward part of a Deep Alignment Network stage. The kernels are described as height × width × depth, stride.

Name	Shape-in	Shape-out	Kernel
conv1a	112×112×1	112×112×64	3×3×1,1
conv1b	112×112×64	112×112×64	3×3×64,1
pool1	112×112×64	56×56×64	2×2×1,2
conv2a	56×56×64	56×56×128	3×3×64,1
conv2b	56×56×128	56×56×128	3×3×128,1
pool2	56×56×128	28×28×128	2×2×1,2
conv3a	28×28×128	28×28×256	3×3×128,1
conv3b	28×28×256	28×28×256	3×3×256,1
pool3	28×28×256	14×14×256	2×2×1,2
conv4a	14×14×256	14×14×512	3×3×256,1
conv4b	14×14×512	14×14×512	3×3×512,1
pool4	14×14×512	7×7×512	2×2×1,2
fc1	7×7×512	1×1×256	-
fc2	1×1×256	1×1×136	-

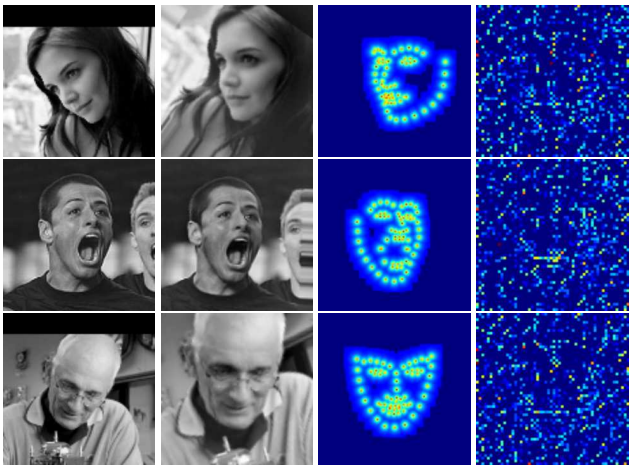

Figure 3. Selected images from the IBUG dataset and intermediate results after 1 stage of DAN. The columns show: the input image I, the input image normalized to canonical shape using transform T_2, the landmark heatmap showing $T_2(S_1)$, the corresponding feature image.

the training procedure.

3.1. Feed-forward neural network

The structure of the feed-forward part of each stage is shown in Table 1. With the exception of max pooling layers and the output layer, every layer takes advantage of batch normalization and uses Rectified Linear Units (ReLU) for activations. A dropout [27] layer is added before the first fully connected layer. The last layer outputs the update ΔS_t to the current estimate of the landmark positions.

The overall shape of the feed-forwad network was inspired by the network used in [26] for the ImageNet ILSVRC 2014 competition.

3.2. Normalization to canonical shape

In DAN the input image I is transformed for each stage so that the current estimates of the landmarks are aligned with the canonical face shape S_0. This normalization step allows the further stages of DAN to be invariant to a given family of transforms. This in turn simplifies the alignment task and improves accuracy.

The Transform Estimation layer of our network is responsible for estimating the parameters of transform T_{t+1} at the output of stage t. As input the layer takes the output of the current stage S_t. Once T_{t+1} is estimated the Image Transform and the Landmark Transform layers transform the image I and landmarks S_t to the canonical pose. The image is transformed using bilinear interpolation. Note that for the first stage of DAN the normalization step is not necessary since the input shape is always the average face shape S_0, which is also the canonical face shape.

Since the input image is transformed, the output of every stage has to be transformed back to match the original image, the output of any DAN stage is thus:

$$S_t = T_t^{-1}(T_t(S_{t-1}) + \Delta S_t), \tag{2}$$

where ΔS_t is the output of the last layer of stage t and T_t^{-1} is the inverse of transform T_t

A similar normalization step has been previously proposed in [31] with the use of affine transforms. In our implementation we chose to use similarity transforms as they do not cause non-uniform scaling and skewing of the output image. Figure 3 shows examples of images before and after the transformation.

3.3. Landmark heatmap

The landmark heatmap is an image where the intensity is highest in the locations of landmarks and it decreases with the distance to the closest landmark. Thanks to the use of landmark heatmaps the Convolutional Neural Network can infer the landmark locations estimated by the previous stage. In consequence DAN can perform face alignment based on entire facial images.

At the input to a DAN stage the landmark heatmap is created based on the landmark estimates produced by the previous stage and transformed to the canonical pose: $T_t(S_{t-1})$. The heatmap is generated using the following equation:

$$H(x,y) = \frac{1}{1 + \min_{s_i \in T_t(S_{t-1})} ||(x,y) - s_i||}, \tag{3}$$

where H is the heatmap image and s_i is the i-th landmark of $T_t(S_{t-1})$. In our implementation the heatmap values are only calculated in a circle of radius 16 around each landmark to improve performance. Note that similarly to normalization, this step is not necessary at the input of the first

stage, since the input shape is always assumed to be S_0, which would result in an identical heatmap for any input.

An example of a face image and a corresponding landmark heatmap is shown in Figure 3.

3.4. Feature image layer

The feature image layer F_t is an image created from a dense layer connected to the fc1 layer (see Table 1) of the previous stage $t-1$. Such a connection allows any information learned by the preceding stage to be transferred to the consecutive stage. This naturally complements the heatmap which transfers the knowledge about landmark locations learned by the previous stage.

The feature image layer is a dense layer which has 3136 units with ReLU activations. The output of this dense layer is reshaped to a 56×56 2D layer and upscaled to 112×112, which is the input shape of DAN stages. We use the smaller 56×56 image rather than 112×112 since it showed similar results in our experiments, with considerably less parameters. Figure 3 shows an example of a feature image.

3.5. Training procedure

The stages of DAN are trained sequentially. The first stage is trained by itself until the validation error stops improving. Subsequently the connection layers and the second stage are added and trained. This procedure is repeated until further stages stop reducing the validation error.

While many face alignment methods [4, 33, 22] learn a model that minimizes the Sum of Squared Errors of the landmark locations, DAN minimizes the landmark location error normalized by the distance between the pupils:

$$\min_{\Delta S_t} \frac{||T_t^{-1}(T_t(S_{t-1}) + \Delta S_t) - S^*||}{d_{ipd}}, \tag{4}$$

where S^* is a vector of ground truth landmark locations, T_t is the transform that normalizes the input image and shape for stage t and d_{ipd} is the distance between the pupils of S^*. The use of this error is motivated by the fact that it is a far more common [33, 22, 36] benchmark for face alignment methods than the Sum of Squared Errors.

Thanks to the fact that all of the layers used in DAN are differentiable DAN can also be trained end-to-end. In order to evaluate end-to-end training in DAN we have experimented with several approaches. Pre-training the first stage for several epochs followed by training of the entire network yielded similar accuracy to the proposed approach but the training was significantly longer. Training the entire network from scratch yielded results significantly inferior to the proposed approach.

While we did not manage to obtain improved results with end-to-end training we believe that it is possible with a better training strategy. We leave the creation of such a strategy for future work.

4. Experiments

In this section we perform an extensive evaluation of the proposed method on several public datasets as well as the test set of the Menpo challenge [34] to which we have submitted our method. The following paragraphs detail the datasets, error measures and implementation. Section 4.1 compares our method with the state of the art, while section 4.2 shows our results in the Menpo challenge. Section 4.3 discusses the influence of the number of stages on the performance of DAN.

Datasets In order to evaluate our method we perform experiments on the data released for the 300W competition [23] and the recently introduced Menpo challenge [34] dataset.

The 300W competition data is a compilation of images from five datasets: LFPW [2], HELEN [18], AFW [37], IBUG [23] and 300W private test set [23]. The last dataset was originally used for evaluating competition entries and at that time was private to the organizers of the competition, hence the name. Each image in the dataset is annotated with 68 landmarks [24] and accompanied by a bounding box generated by a face detector. We follow the most established approach [22, 36, 17, 32] and divide the 300-W competition data into training and testing parts. The training part consists of the AFW dataset as well as training subsets of LFPW and HELEN, which results in a total of 3148 images. The test data consists of the remaining datasets: IBUG, 300W private test set, test sets of LFPW, HELEN. In order to facilitate comparison with previous methods we split this test data into four subsets:

- the *common subset* which consists of the test subsets of LFPW and HELEN (554 images),
- the *challenging subset* which consists of the IBUG dataset (135 images),
- the *300W public test set* which consists of the test subsets of LFPW and HELEN as well as the IBUG dataset (689 images),
- the *300W private test set* (600 images).

The annotation for the images in the 300W public test set were originally published for the 300W competition as part of its training set. We use them for testing as it became a common practice to do so in the recent years [22, 36, 32, 19, 17].

The Menpo challenge dataset consists of semi-frontal and profile face image datasets. In our experiments we only use the semi-frontal dataset. The dataset consists of training and testing subsets containing 6679 and 5335 images respectively. The training subset consists of images from

the FDDB [13] and AFLW [16] datasets. The image were annotated with the same set of 68 landmarks as the 300W competition data but no face detector bounding boxes. The annotations of the test subset have not been released.

Error measures Several measures of face alignment error for an individual face image have been recently introduced:

- the mean distance between the localized landmarks and the ground truth landmarks divided by the inter-ocular distance (the distance between the outer eye corners) [36, 22, 32],
- the mean distance between the localized landmarks and the ground truth landmarks divided by the inter-pupil distance (the distance between the eye centers) [30, 23],
- the mean distance between the localized landmarks and the ground truth landmarks divided by the diagonal of the bounding box [34].

In our work, we report our results using all of the above measures. For evaluating our method on the test datasets we use three metrics: the mean error, the area under the cumulative error distribution curve (AUC_α) and the failure rate.

Similarly to [31, 30], we calculate AUC_α as the area under the cumulative distribution curve calculated up to a threshold α, then divided by that threshold. As a result the range of the AUC_α values is always 0 to 1. Following [30], we consider each image with an inter-ocular normalized error of 0.08 or greater as failure and use the same threshold for $AUC_{0.08}$. In all the experiments we test on the full set of 68 landmarks.

Implementation We train two models, DAN which is trained on the training subset of the 300W competition data and DAN-Menpo which is trained on both the above mentioned dataset and the Menpo challenge training set. Data augmentation is performed by mirroring around the Y axis as well as random translation, rotation and scaling, all sampled from normal distributions. During data augmentation a total of 10 images are created from each input image in the training set.

Both models (DAN and DAN-Menpo) consist of two stages. Training is performed using Theano 0.9.0 [29] and Lasagne 0.2 [9]. For optimization we use Adam stochastic optimization [15] with an initial step size of 0.001 and mini batch size of 64. For validation we use a random subset of 100 images from the training set.

The Python implementation runs at 73 fps for images processed in parallel and at 45 fps for images processed sequentially on a GeForce GTX 1070 GPU. We believe that the processing speed can be further improved by optimizing the implementation of some of our custom layers, most notably the Image Transform layer.

To enable reproducible research, we release the source code of our implementation as well as the models used in the experiments[1]. The published implementation also contains an example of face tracking with the proposed method.

4.1. Comparison with state-of-the-art

We compare the DAN model with state-of-the-art methods on all of the test sets of the 300W competition data. We also show results for the DAN-Menpo model but do not perform comparison since at the moment there are no published methods that use this dataset for training. For each test set we initialize our method using the face detector bounding boxes provided with the datasets.

Tables 2 and 3 show the mean error, $AUC_{0.08}$ and the failure rate of the proposed method and other methods on the *300W public test set*. Table 4 shows the mean error, the $AUC_{0.08}$ and failure rate on the *300W private test set*.

All of the experiments performed on the two most difficult test subsets (the *challenging subset* and the *300W private test set*) show state-of-the-art results, including:

- a failure rate reduction of 60% on the *300W private test set*,
- a failure rate reduction of 72% on the *300W public test set*,
- a 9% improvement of the mean error on the *challenging subset*.

This shows that the proposed DAN is particularly suited for handling difficult face images with a high degree of occlusion and variation in pose and illumination.

4.2. Results on the Menpo challenge test set

In order to evaluate the proposed method on the Menpo challenge test dataset we have submitted our results to the challenge and received the error scores from the challenge organizers. The Menpo test data differs from the other datasets we used in that it does not include any bounding boxes which could be used to initialize face alignment. For that reason we have decided to use a two step face alignment procedure, where the first step serves as an initialization for the second step.

The first step performs face alignment using a square initialization bounding box placed in the middle of the image with a size set to a percentage of image height. The second step takes the result of the first step, transforms the landmarks and the image to the canonical face shape and

[1] https://github.com/MarekKowalski/DeepAlignmentNetwork

Table 2. Mean error of face alignment methods on the 300W public test set and its subsets. All values are shown as percentage of the normalization metric.

Method	Common subset	Challenging subset	Full set
inter-pupil normalization			
ESR [4]	5.28	17.00	7.58
SDM [33]	5.60	15.40	7.52
LBF [22]	4.95	11.98	6.32
cGPRT [19]	-	-	5.71
CFSS [36]	4.73	9.98	5.76
Kowalski et al. [17]	4.62	9.48	5.57
RAR [32]	**4.12**	8.35	4.94
DAN	4.42	**7.57**	5.03
DAN-Menpo	4.29	**7.05**	**4.83**
inter-ocular normalization			
MDM [30]	-	-	4.05
Kowalski et al. [17]	3.34	6.56	3.97
DAN	**3.19**	**5.24**	**3.59**
DAN-Menpo	**3.09**	**4.88**	**3.44**
bounding box diagonal normalization			
DAN	**1.35**	**2.00**	**1.48**
DAN-Menpo	**1.31**	**1.87**	**1.42**

Table 3. AUC and failure rate of face alignment methods on the 300W public test set.

Method	$AUC_{0.08}$	Failure (%)
inter-ocular normalization		
ESR [4]	43.12	10.45
SDM [33]	42.94	10.89
CFSS [36]	49.87	5.08
MDM [30]	52.12	4.21
DAN	**55.33**	**1.16**
DAN-Menpo	**57.07**	**0.58**

Table 4. Results of face alignment methods on the 300W private test set. Mean error is shown as percentage of the inter-ocular distance.

Method	Mean error	$AUC_{0.08}$	Failure (%)
inter-ocular normalization			
ESR [4]	-	32.35	17.00
CFSS [36]	-	39.81	12.30
MDM [30]	5.05	45.32	6.80
DAN	**4.30**	**47.00**	**2.67**
DAN-Menpo	**3.97**	**50.84**	**1.83**

creates a bounding box around the transformed landmarks. The transformed image and bounding box are used as input to face alignment. An inverse transform is later applied to get landmark coordinates for the original image.

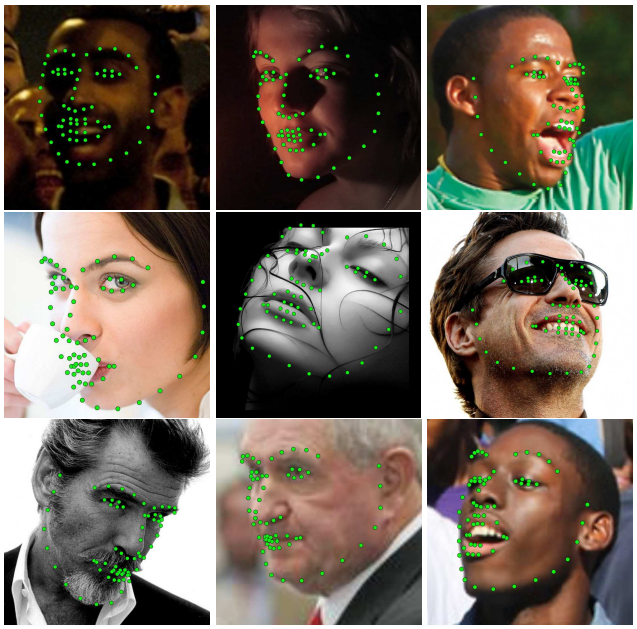

Figure 4. The 9 *worst* results on the challenging subset (IBUG dataset) in terms of inter-ocular error produced by the DAN model. Only the first 7 images have an error of more than 0.08 inter-ocular distance and can be considered failures.

In order to determine the optimal size of the bounding boxes in the first step we ran DAN on a small subset of the Menpo test set for several bounding box sizes. The optimal size was determined using a method that would estimate the face alignment error of a given set of landmarks and an image. Said method extracts HOG [8] features at each of the landmarks and uses a linear model to estimate the error. The method was trained on the 300W training set using ridge regression. The chosen bounding box size was 46% of the image height.

Figure 6 and Table 5 show the CED curve, mean error, $AUC_{0.03}$ and failure rate for the DAN-Menpo model on the Menpo test set. In all cases the errors are calculated using the diagonal of the bounding box normalization, used by the challenge organizers. For the AUC and the failure rate we have chosen a threshold of 0.03 of the bounding box diagonal as it is approximately equivalent to 0.08 of the interocular distance used in the previous chapter.

Figure 5 shows examples of images from the Menpo test set and corresponding results produced by our method. Note that even though DAN was trained primarily on semi-frontal images it can handle fully profile images as well.

4.3. Further evaluation

In this subsection we evaluate several DAN models with a varying number of stages on the 300W private test set. All of the models were trained identically to the DAN model from section 4.1. Table 6 shows the results of our eval-

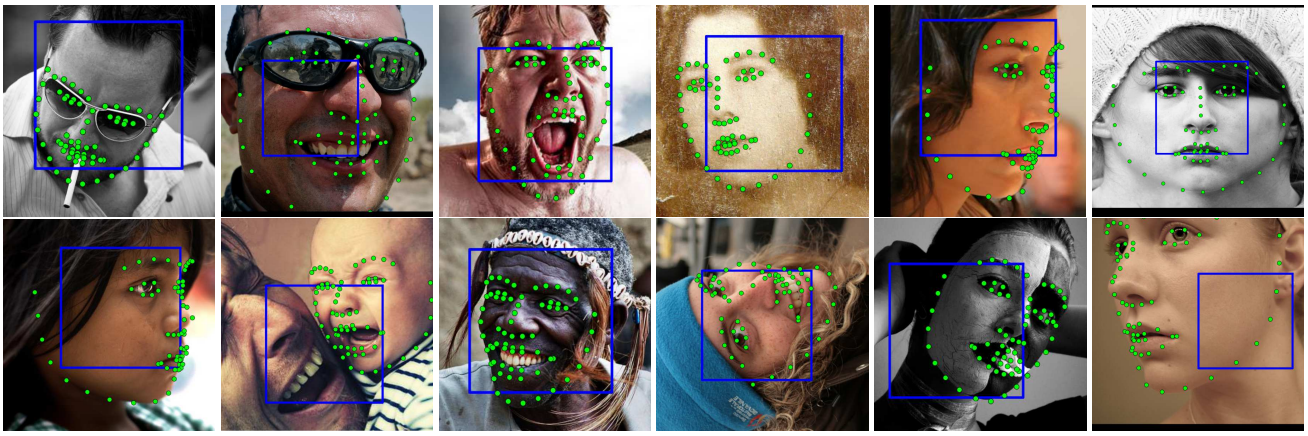

Figure 5. Results of our submission to the Menpo challenge on some of the difficult images of the Menpo test set. The blue squares denote the initialization bounding boxes. The images were cropped to better visualize the results, in the original images the bounding boxes are always located in the center.

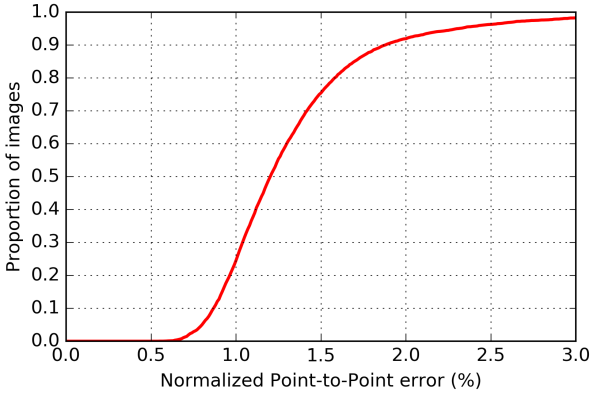

Figure 6. The Cumulative Error Distribution curve for the DAN-Menpo model on the Menpo test set. The Point-to-Point error is shown as percentage of the bounding box diagonal.

Table 5. Results of the proposed method on the semi-frontal subset of the Menpo test set. Mean error is shown as percentage of the bounding box diagonal.

Method	Mean error	$AUC_{0.03}$	Failure (%)
bounding box diagonal normalization			
DAN-Menpo	**1.38**	**56.20**	**1.74**

uation. The addition of the second stage increases the $AUC_{0.08}$ by 20% while the mean error and failure rate are reduced by 14% and 56% respectively. The addition of a third stage does not bring significant benefit in any of the metrics.

5. Conclusions

In this paper, we introduced the Deep Alignment Network - a robust face alignment method based on convolutional neural networks. Contrary to the recently proposed face alignment methods, DAN performs face alignment based on entire face images, which makes it highly robust to large variations in both initialization and head pose. Using entire face images instead of local patches extracted around landmarks is possible thanks to the use of novel landmark heatmaps which transmit the information about landmark locations between DAN stages. Extensive evaluation performed on two challenging, publicly available datasets shows that the proposed method improves the state-of-the-art failure rate by a significant margin of over 70%.

Future research includes investigation of new strategies for training DAN in an end-to-end manner. We also plan to introduce learning into the estimation of the transform T_t that normalizes the shapes and images between stages.

Table 6. Results of the proposed method with a varying number of stages on the 300W private test set. Mean error is shown as percentage of the inter-ocular distance.

# of stages	Mean error	$AUC_{0.08}$	Failure (%)
inter-ocular normalization			
1	5.02	39.04	6.17
2	4.30	47.00	2.67
3	4.32	47.08	2.67

6. Acknowledgements

The work presented in this article was supported by The National Centre for Research and Development grant number DOB-BIO7/18/02/2015. The results obtained in this work were a basis for developing the software for the grant sponsor, which is different from the software published with this paper.

We thank NVIDIA for donating a Titan X Pascal GPU which was used to train the proposed neural network.

References

[1] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Robust discriminative response map fitting with constrained local models. In *Computer Vision and Pattern Recognition, 2013 IEEE Conference on*, pages 3444–3451, 2013. 2

[2] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2930–2940, Dec 2013. 5

[3] A. Bulat and G. Tzimiropoulos. Two-stage convolutional part heatmap regression for the 1st 3d face alignment in the wild (3dfaw) challenge. In *Computer Vision Workshop, 14th European Conference on*, 2016. 2, 3

[4] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. *International Journal of Computer Vision*, 107(2):177–190, Apr. 2014. 1, 2, 5, 7

[5] 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. 2

[6] P. M. Corcoran, F. Nanu, S. Petrescu, and P. Bigioi. Real-time eye gaze tracking for gaming design and consumer electronics systems. *IEEE Transactions on Consumer Electronics*, 58(2):347–355, May 2012. 1

[7] D. Cristinacce and T. F. Cootes. Feature detection and tracking with constrained local models. In *British Machine Vision Conference*, volume 1, page 3, 2006. 2

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005 IEEE Conference on*, volume 1, pages 886–893 vol. 1, June 2005. 7

[9] S. Dieleman, J. Schlter, C. Raffel, E. Olson, S. K. Snderby, D. Nouri, et al. Lasagne: First release., Aug. 2015. 6

[10] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, V. Vapnik, et al. Support vector regression machines. *Advances in neural information processing systems*, 9:155–161, 1997. 2

[11] H. Fan and E. Zhou. Approaching human level facial landmark localization by deep learning. *Image and Vision Computing*, 47:27–35, 2016. 2, 3

[12] P. L. Hsieh, C. Ma, J. Yu, and H. Li. Unconstrained realtime facial performance capture. In *Computer Vision and Pattern Recognition, 2015 IEEE Conference on*, pages 1675–1683, June 2015. 1

[13] V. Jain and E. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010. 6

[14] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Computer Vision and Pattern Recognition, 2014 IEEE Conference on*, pages 1867–1874, 2014. 2

[15] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Learning Representations, 3rd International Conference on*, 2014. 6

[16] M. Koestinger, P. Wohlhart, P. M. Roth, and H. Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, 2011. 6

[17] M. Kowalski and J. Naruniec. Face alignment using k-cluster regression forests with weighted splitting. *IEEE Signal Processing Letters*, 23(11):1567–1571, Nov 2016. 2, 5, 7

[18] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang. Interactive facial feature localization. In *Computer Vision, 12th European Conference on*, pages 679–692, 2012. 5

[19] D. Lee, H. Park, and C. D. Yoo. Face alignment using cascade gaussian process regression trees. In *Computer Vision and Pattern Recognition, 2015 IEEE Conference on*, pages 4204–4212, June 2015. 2, 5, 7

[20] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110. 2

[21] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004. 2

[22] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *Computer Vision and Pattern Recognition, 2014 IEEE Conference on*, pages 1685–1692, June 2014. 1, 2, 5, 6, 7

[23] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *Computer Vision and Pattern Recognition Workshops, 2013 IEEE Conference on*, pages 397–403, Dec 2013. 1, 2, 3, 5, 6

[24] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. A semi-automatic methodology for facial landmark annotation. In *Computer Vision and Pattern Recognition Worshops, 2013 IEEE Conference on*, pages 896–903, 2013. 3, 5

[25] E. Sariyanidi, H. Gunes, and A. Cavallaro. Automatic analysis of facial affect: A survey of registration, representation, and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(6):1113–1133, June 2015. 1

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *Computing Research Repository*, abs/1409.1556, 2014. 4

[27] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 4

[28] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Computer Vision and Pattern Recognition, 2014 IEEE Conference on*, pages 1701–1708, 2014. 1

[29] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016. 6

[30] G. Trigeorgis, P. Snape, M. Nicolaou, and E. Antonakos. Mnemonic descent method: A recurrent process applied for end-to-end face alignment. In *Computer Vision and Pattern Recognition, 2016 IEEE Conference on*, 2016. 1, 2, 3, 6, 7

[31] O. Tuzel, T. K. Marks, and S. Tambe. *Robust Face Alignment Using a Mixture of Invariant Experts*, pages 825–841. Springer International Publishing, Cham, 2016. 2, 4, 6

[32] S. Xiao, J. Feng, J. Xing, H. Lai, S. Yan, and A. Kassim. *Robust Facial Landmark Detection via Recurrent Attentive-Refinement Networks*, pages 57–72. Springer International Publishing, Cham, 2016. 2, 5, 6, 7

[33] X. Xiong and F. D. la Torre. Supervised descent method and its applications to face alignment. In *Computer Vision and Pattern Recognition, 2013 IEEE Conference on*, pages 532–539, June 2013. 1, 2, 5, 7

[34] S. Zafeiriou, G. Trigeorgis, G. Chrysos, J. Deng, and J. Shen. The menpo facial landmark localisation challenge: A step closer to the solution. In *Computer Vision and Pattern Recognition, 1st Faces in-the-wild Workshop-Challenge, 2017 IEEE Conference on*, 2017. 5, 6

[35] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin. Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In *Computer Vision and Pattern Recognition Workshops, 2013 IEEE Conference on*, pages 386–391, 2013. 2, 3

[36] S. Zhu, C. Li, C. C. Loy, and X. Tang. Face alignment by coarse-to-fine shape searching. In *Computer Vision and Pattern Recognition, 2015 IEEE Conference on*, pages 4998–5006, June 2015. 1, 2, 5, 6, 7

[37] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition, 2012 IEEE Conference on*, pages 2879–2886, 2012. 5