Facial Landmark Detection with Tweaked Convolutional Neural Networks

Yue Wu,* Tal Hassner,* KangGeon Kim, Gérard Medioni, and Prem Natarajan

Abstract—This paper concerns the problem of facial landmark detection. We provide a unique new analysis of the features produced at intermediate layers of a convolutional neural network (CNN) trained to regress facial landmark coordinates. This analysis shows that while being processed by the CNN, face images can be partitioned in an unsupervised manner into subsets containing faces in similar poses (i.e., 3D views) and facial properties (e.g., presence or absence of eye-wear). Based on this finding, we describe a novel CNN architecture, specialized to regress the facial landmark coordinates of faces in specific poses and appearances. To address the shortage of training data, particularly in extreme profile poses, we additionally present data augmentation techniques designed to provide sufficient training examples for each of these specialized sub-networks. The proposed *Tweaked CNN* (TCNN) architecture is shown to outperform existing landmark detection methods in an extensive battery of tests on the AFW, ALFW, and 300W benchmarks. Finally, to promote reproducibility of our results, we make code and trained models publicly available through our project webpage.

Index Terms—I.5.4.d Face and gesture recognition; I.5.1.d Neural nets

1 Introduction

Facial landmark detection plays a key role in many face processing applications, including head pose estimation [1], [2], emotion classification [3], face alignment in 2D [4], [5] and 3D (e.g., *frontalization* [6]) and, of course, face recognition (see, e.g., Sun et al. [7] and many others). This task is particularly daunting considering the real-world, unconstrained imaging conditions typically assumed: images often portray faces in myriads of poses, expressions, occlusions and more, any of which can affect landmark appearances, locations or even presence.

Many effective methods were proposed to handle these challenges. Several use classifiers and robust representations to search for specific facial parts, further disambiguating detections by constraining landmark arrangements [2], [8], [9]. Some regress detections directly [4], [5], [10], [11]. These methods are known to be limited in the pose variations they can account for, leading to pose estimation prior to detection [12]. Others are further motivated by specific appearance variations (e.g., pose [2], [9] or occlusions [10], [11]); hence, their performance may not carry over to appearance variations unaccounted for in their design.

In line with the recent success of deep learning, a number of such methods were proposed for regressing landmark coordinates [13], [14], [15], [16], [17]. These methods naturally learn landmark appearance and location variations from huge training sets. To do so, they learn multiple part models [13], [14], hierarchical representations [13], [15] or infuse networks with additional attribute labels [16], [17].

We present a novel convolutional neural network (CNN) for face landmark regression. Contrary to others, our design *does not involve multiple part models*; it is *naturally hierarchical* and requires *no auxiliary labels* beyond landmarks. Our approach partitions the landmark detection problem into sub-domains in an unsupervised manner. Each sub-domain represents faces with similar pose and appearance attributes, determined here without manual labeling. The network is then *tweaked* to specialize itself to each domain.

We are motivated by recent reports that features from intermediate layers of deep networks become progressively task-specific at deeper layers [18], [19]. As we later show, when trained to detect facial landmarks, these specialized features actually reflect head pose and other facial appearance attributes quite well. Fig. 1 illustrates this, showing 64 cluster centers – averages of cluster images – comparing clusters computed using input RGB values (top) vs. input features from the first dense layer of face landmark regression CNN (Sec. 2.1). RGB clusters clearly reflect image intensities. By comparison, intermediate feature clusters appear to contain well-aligned faces with similar poses and facial properties, apparent by their sharp average faces.

We leverage these findings and introduce our *Tweaked CNN* (TCNN). TCNN automatically diverts deep features to separate specialized processing. Tweaking implies fine-tuning the final layers for particular head poses. Thus, heavy processing at the early, convolutional layers are shared among different tweaking branches; differential, specialized treatment is applied only in the final layers. We explain how overfitting is avoided when fine-tuning the final layers, despite limited training data, using a novel alignment-sensitive augmentation technique. Finally, we show that TCNN outperforms existing state of the art, with an efficient architecture and fewer labels required for training.

Contributions. (1) A first analysis of representations produced at intermediate layers of a deep CNN trained for landmark detection, showing them to be surprisingly good at representing different head poses and (some) facial attributes (Sec. 2.2). **(2)** Our TCNN model which improves CNN-based landmark detection by differently tweaking the

- *Y. Wu, T. Hassner and P. Natarajan are with the Information Sciences Institute, USC, CA, USA. E-mail: {yue_wu,hassner,pnataraj}@isi.edu*
- *K. Kim and G. Medioni are with the Institute for Robotics and Intelligent Systems, USC, CA, USA. E-mail: {kanggeok,medioni}@usc.edu*

* *Denotes equal contribution.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2787130, IEEE Transactions on Pattern Analysis and Machine Intelligence

Fig. 1. *Average images for 64 face clusters.* Top: clusters computed using RBG values. These appear misaligned (blurry) and strongly influenced by intensities. Bottom: Images clustered using features from an intermediate layer of a network trained to regress facial landmarks. These are clearly better aligned. We leverage this to tweak network processing based on intermediate representations. (Note: Both results were produced using the same image set.)

processing of different intermediate features (Sec. 3.1). **(3)** A novel data augmentation method which inflates available training data for fine-tuning on different head poses (Sec. 3.2). The benefits of these are demonstrated by reporting extensive facial landmark detection tests on the AFLW, AFW, and 300W benchmarks.

2 Motivation and Supporting Analysis

We begin by studying a *vanilla CNN* trained to detect facial landmarks. Our goal is to "pop the hood" off the CNN to better understand its internal representations and the nature of the information they encode. This analysis provides quantitative support to our novel network design.

2.1 Vanilla network design

Our vanilla CNN is loosely based on a state-of-the-art network for facial landmark coordinate regression [16]. This model was selected for its simplicity as well as to allow direct comparison of our TCNN model with previous work. Its design is illustrated in Fig. 2 (a).[1]

1. Trained network and code available from the project page at www.openu.ac.il/home/hassner/projects/tcnn_landmarks.

Images are processed by a face detector [20], returning bounding box coordinates for each face. Bounding boxes are scaled to 40×40 pixels and represented using RGB values, normalized by subtracting the training set mean image and dividing by its standard deviation.

The network consists of four convolutional layers (denoted $CL_1 \dots CL_4$) with intermittent max pooling layers (*stride*=2). These are followed by a fully connected (dense) layer, FC_5, which is then fully connected to an output with $2 \times m$ values for the m landmark coordinates: $\mathbf{P} = (\mathbf{p}_1, \dots, \mathbf{p}_m) = (x_1, y_1, \dots, x_m, y_m)$. In our experiments, networks are trained to detect $m = 5$ landmarks in the bounding box coordinate frames.

Following the network used by Zhang et al. [16], the activation function after every convolutional layer and every dense layer are absolute hyperbolic tangent, and hyperbolic tangent, respectively. Finally, unlike Zhang et al., we use L2, normalized by the inter-ocular distance as the network loss:

$$\mathcal{L}(\mathbf{P}_i, \hat{\mathbf{P}}_i) = \frac{\|\mathbf{P}_i - \hat{\mathbf{P}}_i\|_2^2}{\|\hat{\mathbf{p}}_{i,1} - \hat{\mathbf{p}}_{i,2}\|_2^2}, \tag{1}$$

where $\mathbf{P}_i$ is the $2 \times m$ vector of predicted coordinates for training image I_i, $\hat{\mathbf{P}}_i$ their ground truth locations, and $\hat{\mathbf{p}}_{i,1}, \hat{\mathbf{p}}_{i,2}$ the reference eye positions. We chose this loss to reflect the standard measure for detection accuracy (Sec. 5).

Training used the *Adam* optimization [21] with the same images and landmarks used by others [14], [16]. This set contains 5,590 images from the LFW collection [22] and 7,876 images from throughout the web, all labeled for five facial landmarks. Whenever the face detector failed to locate a face, we discarded that image from the training/validation set, further counting such test images as failures when reporting performance in Sec. 5. Remaining faces were randomly partitioned, taking 90% (7,571 images) for training and $1,972$ for validation. As we later report, this network performs comparably with the state of the art – despite its straightforward design – possibly due to our use of a loss function more suitable to facial landmark detection.

2.2 What does the network learn?

Once trained, we use the vanilla CNN to extract representations from the input of each layer (including the first – the input RGB values). We analyze these features seeking answers to the following questions:

Q1 Do similar features reflect similar facial properties?
Q2 If so, what are these properties?
Q3 When (at what layer) are these properties captured?

A positive answer to **Q1** suggests that prior to landmark detection, the network internally aggregates particular faces together. Answering **Q2** would provide a relationship between specific facial properties and the values of intermediate features. Finally, **Q3** seeks the layer at which these facial properties naturally emerge. At this layer we can assume knowledge of the facial properties and decide if we can apply specialized (tweaked) treatment to the image, based on these properties.

If intermediate features with similar values indeed capture similar facial properties, then we expect to see features

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2787130, IEEE Transactions on Pattern Analysis and Machine Intelligence

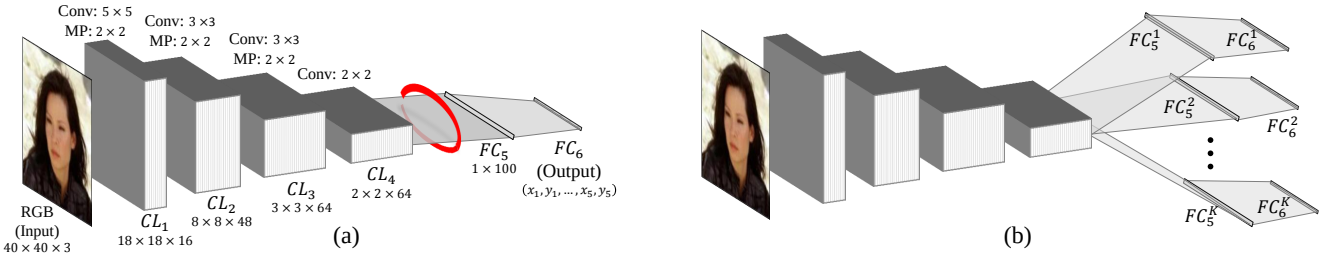


Fig. 2. *CNN architectures.* (a) The vanilla network described in Sec. 2.1 for facial landmark regression. We show that representations extracted from the input to FC_5 (marked in red) are highly specialized and, in particular, reflect head pose and some facial properties. (b) Our Tweaked CNN design, diverting intermediate features to K different subsequent, fine-tuned processes in the same dimensions as the original layers.

TABLE 1
Cluster statistics. Reports median(S_l) ± SD(S_l), where $S_l = \{|C_{l,k}||k = 1..64\}$ is the set containing image numbers in the clusters of layer l.

CL_1	CL_2	CL_3	CL_4	FC_5
92.0 ± 98.3	110.5 ± 55.1	116.0 ± 70.0	118.5 ± 81.6	110.0 ± 58.3

aggregating together into clusters according to these properties. We thus proceed by applying unsupervised clustering to the features of each layer.

Clustering intermediate features. Let $\mathbf{f}_{i,l} = f(I_i, L_l)$ denote the feature vector extracted from the input to layer L_l of our vanilla CNN for training image I_i, where $L_l \in \{CL_1, \ldots, CL_4, FC_5\}$ is the layer name. For n images, we partition the set $\{\mathbf{f}_{i,l}|i = 1, \ldots, n\}_{l\in 1\ldots 5}$ into K clusters, $C_{l,k\in 1,\ldots,K}$, using EM to compute Gaussian mixture models (GMM) [23] and L2 as the feature dissimilarity (corresponding with the normalized L2 used in our cost function, Eq. 1). The vector $\mathbf{f}_{i,l}$ is associated with the cluster with the highest posterior probability:

$$\mathbf{f}_{i,l} \in C_{l,k} \text{ iff } k = \arg\max_k p(C_{l,k}|\mathbf{f}_{i,l}) \quad (2)$$

This analysis uses $K = 64$ clusters per layer. Some per-layer statistics for these clusters are reported in Table 1. Note that GMM provided better results than k-means and was therefore used in all our tests.

Do different clusters of features reflect different landmark positions? Fig. 1 (bottom) already hints at the answer: clusters of features extracted from FC_5 appear to contain aligned images, implying that this is indeed the case.

We analyze this quantitatively using the training set images, I_i, and their ground truth landmarks $\hat{\mathbf{p}}_{i,j\in 1\ldots m} = (x_{i,j}, y_{i,j})$. For each layer L_l and each cluster $C_{l,k}$, we measure the variance $\lambda_{l,k,j}$ along the principle axis of each set of 2D points, $\{\hat{\mathbf{p}}_{i,j}|\mathbf{f}_{i,l} \in C_{l,k}\}_{l,k,j}$. These are then averaged for all clusters in each layer:

$$\mu^P_{l,j} = \frac{1}{K}\sum_{k=1}^{K} \lambda_{l,k,j}. \quad (3)$$

Fig. 3 reports these values for the five landmarks, over the layers L_l, along with their standard errors. The average intra-cluster location variances *drop by half* from the input to FC_5. This is remarkable, as it can be interpreted as suggesting that the network naturally performs hierarchical coarse-to-fine feature localization, where deeper layers better represent landmark positions.

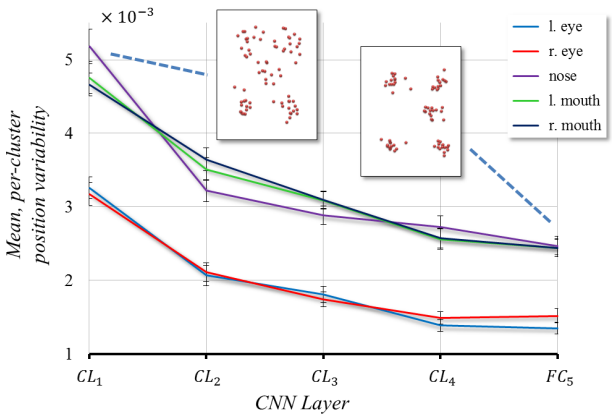


Fig. 3. *Landmark position variability at different layers.* $\lambda_{l,k,j}$ averaged (± SE) over K clusters in each layer (Eq. 3). Scatter plots for the ground truth landmarks of 15 faces from the most variable clusters of RGB (left) and FC_5 (right) layer features are shown, visualizing the reduced variability at the deeper layer.

Facial attributes. Facial landmark positions and facial attributes are very much related. In the past, this connection was mainly utilized going from landmark positions to attribute prediction [24], [1], [3], [25]. To our knowledge, the reverse – using attributes to predict landmarks – was only recently proposed in a system which required manual attribute labels [16], [17].

We measure the relation between clusters and facial attributes using the following binary attribute labels [14], [16]: *male/female*, *smiling/not-smiling* and *yes/no wearing eyeglasses*. The variance of $1/0$ values for an attribute in cluster $C_{l,k}$ is denoted by $\sigma^2_{l,k,a}$ (a indexing the three attributes). A low value of $\sigma^2_{l,k,a}$ reflects a label's uniformity in a cluster. We average these variances over all clusters in each layer:

$$\mu^A_{l,a} = \frac{1}{K}\sum_{k=1}^{K} \sigma^2_{l,k,a}. \quad (4)$$

These values are reported in Fig. 4. Clusters from deeper layers appear far less variable in smiling/not-smiling faces. Due to the small number of positive eyeglass attributes in the data (15.3%, compared with 57.2% for smiling), this attribute is less varied to begin with, but becomes far less so in deeper layer clusters. The results are expected: landmark positions and appearances are heavily influenced by these two attributes. Gender, however, seems to become more varied in higher layers. This is surprising, as gender was very beneficial to detection in the work of Zhang et al. [16].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2787130, IEEE Transactions on Pattern Analysis and Machine Intelligence

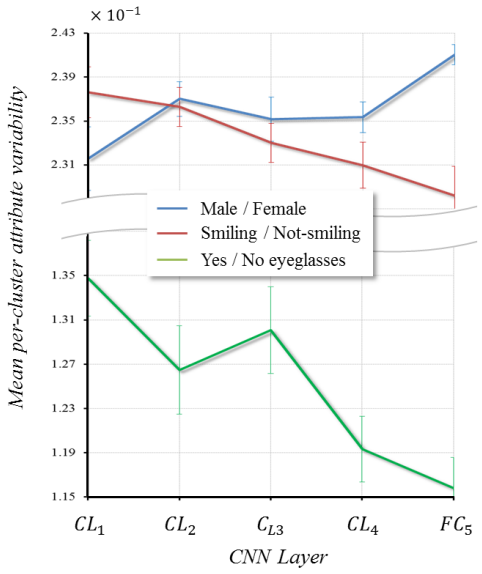


Fig. 4. *Variance of attribute labels at different layers.* $\sigma^2_{l,k,a}$ averaged ($\pm$ SE) over K clusters in each layer (Eq. 4).

A possible explanation is that in the low resolution, 40×40 pixel images used here, landmark position and appearance differences between genders are not substantial enough to aid the CNN in detection. Finally, these results also indicate that clustering images based solely on pose, without considering other facial attributes [12] may not be optimal.

2.3 Discussion: What can we learn from this?

These findings show that when trained to regress facial landmark positions, a network internally learns representations which discriminate between facial appearances and poses. Specifically, at the input to the first fully connected layer (represented by a red band in Fig. 2 (a)), *we know rough head pose*. With this information we can train *pose-specific* landmark regressors. In the past, others proposed similar two-step approaches – beginning with rough localization and then using it to accurately detect landmarks [12], [13], [15], [16], [26], [27]. Contrary to these and others, here, rough pose *emerges naturally* from the network and is partitioned into sub-domains in an *unsupervised manner*. As we next show, the final landmark regressors also naturally emerge from the network.

3 The Tweaked CNN Model

Our analysis in Sec. 2.2 is consistent with previous reports on the specialization of late layers in deep networks [18], [19]. To our knowledge, however, the nature of this effect was never previously explored for face images or exploited to improve performance.

We propose to utilize this phenomenon to improve accuracy by fine-tuning multiple versions of the final network layers: each one trained on an automatically determined subset of the original training data, each subset containing faces with similar appearances. We therefore obtain multiple versions of the final layers, each one specialized to specific appearances. To avoid overfitting, Sec. 3.2 further presents an alignment-sensitive data augmentation method.

3.1 Tweaking: fine-tuning to sub-domains

Our TCNN is illustrated in Fig. 2 (b). We begin by training the vanilla CNN for facial landmark regression (Sec. 2.1). Once trained, it is used to extract features from the input to FC_5 and aggregate them into K GMMs using EM (Sec. 2.2). Next, we fine-tune the remaining weights, from FC_5 to the output, separately for each cluster using only its images. Here *early stopping* is used to fine-tune each subnetwork; that is, if validation loss does not improve for 50 epochs, we cease fine-tuning that cluster.

We emphasize that fine-tuning only involves the weights in final layers; previous layers are kept frozen. In addition, fine-tuning the network for each cluster uses the same layer dimensions (weight arrangements), commencing with the vanilla network weights. Thus, TCNN training requires little effort beyond training the initial vanilla CNN. This should be compared with methods that train multiple CNN models for multiple resolutions [15] or parts [13].

In practice, training the vanilla CNN on a dedicated machine required $\sim$6.5 hours. By comparison, tweaking a single cluster took $\sim$2 *minutes*. Of course, tweaking different clusters can be performed on separate machines in parallel, and so the added time for tweaking is very low.

Estimating landmark positions for a query photo I_Q begins by using the vanilla CNN to extract the values from the *input* to FC_5: $\mathbf{f}_{Q,5} = f(I_Q, FC_5)$. It is then assigned to the cluster $C_{5,k}$ which maximizes the posterior probability (Eq. 2). From this point onward, the network proceeds processing this feature vector using only the layers fine-tuned for $C_{5,k}$, finally returning the output from FC^k_6.

Ensuring horizontal symmetry. Our process treats horizontally flipped versions of the same face differently – by different tweaked processes – and one may be better than the other. To ensure that our detector is mirror-invariant, we apply it twice to each image: the original and its horizontally mirrored. Each two predictions are then averaged (after mirroring the coordinates of the flipped image) to obtain final detection.

3.2 Alignment-sensitive data augmentation

The numbers of training images in each cluster, reported in Table 1, are too low to train even the last layers of the network without risking overfitting. This is particularly true for clusters representing extreme profile poses, where few examples are available in the training set. We address this problem by augmenting the training data.

Generic augmentation methods such as *oversampling* [28] and *mirroring* [27] are unsuitable here: each tweaked, fine-tuned network trains on representations from the same cluster. These, as we showed earlier, should all be well aligned. Oversampling and mirroring both introduce misaligned images into each cluster, increase landmark position variability, and thus undermine the goal of our fine-tuning. Ostensibly, we could add data to each cluster by sampling from the GMM components (see Eq. 2). These, however, are defined on intermediate features and cannot provide the ground truth landmark labels required for training.

Instead, we propose augmenting the image set used to fine-tune tweaked layers in an *alignment-sensitive* manner.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2787130, IEEE Transactions on Pattern Analysis and Machine Intelligence

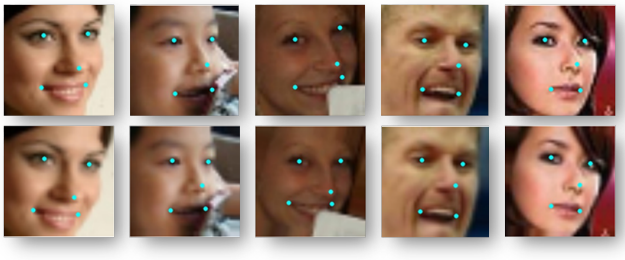


Fig. 5. *Alignment-sensitive data augmentation.* Top: Training images from the same FC_5 cluster. Bottom: Images added to the cluster to increase training set size. Bottom images are noticeably different from their origins in the top, yet remain in their original cluster. Ground truth landmark positions used to align these images appear in cyan.

Let $I_i, I_{i'}$ be two training images, randomly selected as being associated to the same cluster, $C_{5,k}$, and $\hat{\mathbf{p}}_{i,j}, \hat{\mathbf{p}}_{i',j}, j \in 1\ldots,m$ their m respective ground truth landmarks. We estimate the non-reflective similarity transform $\mathbf{H}$ mapping $\hat{\mathbf{p}}_{i',j}$ to $\hat{\mathbf{p}}_{i,j}$ using standard least squares [29] and use this to backwards warp I_i to the coordinate frame of $I_{i'}$; i.e.,

$$I'_i(x,y) \triangleq I_i(\mathbf{H}^{-1}(x,y)). \quad (5)$$

The new image, I'_i is verified to belong to $C_{5,k}$ by extracting its feature representation from the input to FC_5 and associating it with a cluster $C_{5,k'}$ using Eq. 2. If $k \neq k'$, then the generated image does not belong in the same cluster with the two images used to produce it, and it is therefore rejected. In practice, <40% of the generated images failed this test, typically due to artifacts introduced by warping. This should be compared to the over 96% rejected when using images from other clusters. Accepted images I'_i are added to the training along with the landmark labels of $I_{i'}$.

This data augmentation approach can presumably be used with any number of clusters, particularly, when there is only one, to augment the data used to train the vanilla CNN. In such cases, however, the rejection step mentioned above is meaningless, and all generated images are used for training. In practice, training the CNN with this data augmentation technique did not improve results, and so we do not apply it to the single-cluster, vanilla CNN results reported in Sec. 5.

Fig. 5 provides a few examples of images added by this process to our training. These are slightly, but noticeably misaligned with their sources. They therefore introduce variation to each tweaking training set, yet still belong to their original clusters. We use this process to artificially raise the number of training images in each cluster to $5,000$ images. We empirically found smaller numbers of augmented images to result in overfitting. In particular, without augmentation, our data was insufficient to train our network. Hence, all TCNN results reported in this paper were obtained following training with this augmentation. Finally, larger numbers of augmented images were found to provide no meaningful performance gain.

4 Relation to Existing Work

We next compare our TCNN design with relevant existing models and detectors.

Several previous methods proposed fine-tuning late layers to specific data [19], [30], [31]. These, however, focus on domain transfer applications where fine-tuned layers are trained with new supervised data from different problem domains. We, by contrast, improve network performance on augmented subsets of the original data, determined in an unsupervised manner.

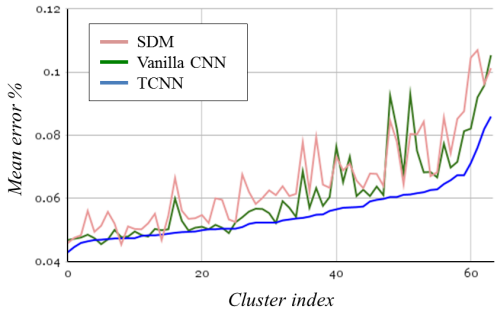


Fig. 6. *Effect of tweaking.* Validation set, per cluster mean error rate for vanilla CNN (green) vs. TCNN (blue) (Sec. 3). SDM [5] provided as reference in pink. $K = 64$ clusters produced from FC_5 features, sorted by TCNN performance. The same order (left-to-right, top-to-bottom) was used for the cluster mean images in Fig. 1 (bottom). Evidently, the biggest improvements were obtained for out-of-plane views.

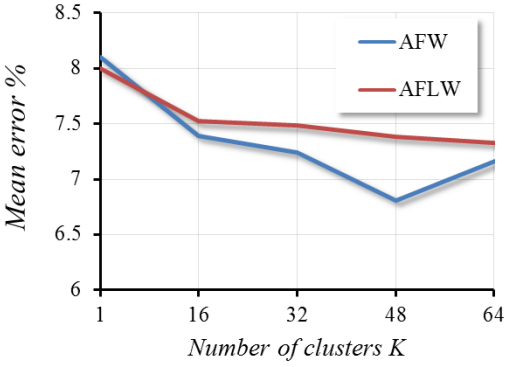


Fig. 7. *Number of clusters vs. mean error.* Results on AFW [2] (blue) and AFLW [33] (red) for TCNN models with different numbers of tweaked processes, K. Lower numbers are better.

TCNN can be considered as operating in a multiscale manner: coarse positions are reflected by intermediate representations and then localized by tweaked, final layers. Multiscale networks were previously proposed, and we refer to Xie and Tu [32] for a survey. None of these, however, use internal representations to guide multiscale processing as we do. As we mention in Sec. 2.3, though some landmark detection methods also take a course-to-fine approach, these are very different from the one proposed here.

Multiscale landmark detection methods include Zhang et al. [15], who train multiple networks for multiple resolutions, or Liang et al. [13] who designed a novel network architecture to determine coarse landmark positions and then localize them in high resolution with landmark-specific subnetworks. Both are very different from our approach.

Existing landmark detectors, including CFSS [26], SDM [5] and many others, partition image space to (typically few) facial poses in a supervised manner, training separate models for each pose. TCNN, on the other hand, does not partition faces by poses and our partition is unsupervised: We treat images differently based on their intermediate feature values which we cluster using unsupervised GMM. Our novel analysis in Sec. 2.2 shows that by doing so, faces with similar shapes and appearances (e.g., poses, yes,

but also expressions, eye-wear and possibly more) cluster together naturally, allowing for tweaked treatment based on appearances, without first manually determining a space partition in a supervised manner.

Finally, the recent work of Zhang et al. [16], [17] detects facial landmarks with a network trained using multitask learning of both landmarks and attributes. To do so, they manually specify facial attributes and label the training images accordingly. We use no such auxiliary labels; instead, following our analysis in Sec. 2.2, we infer relevant poses and attributes in an unsupervised manner. We next show that our TCNN exceeds the performance of the referenced work with a simpler network design.

5 Experiments

5.1 Five-point landmark detection

Evaluation criteria. In our tests we use the detector *error rate* to report accuracy [34]. It is computed by normalizing the mean distance between predicted to ground truth landmark locations to a percent of the inter-ocular distance.

The effect of tweaking by fine-tuning. Fig. 6 demonstrates the effect of tweaking by fine-tuning (Sec. 3.1) with alignment-sensitive data augmentation (Sec. 3.2) on landmark detection accuracy. It reports the per-cluster detection error rate on validation set images, comparing vanilla CNN performance with TCNN. For convenience, cluster labels are sorted by ascending TCNN errors. Apparently, in nearly all clusters, TCNN manages to improve accuracy – in some cases by several percent.

We note that the cluster centers presented in Fig. 1 (bottom) are also sorted (left-to-right, top-to-bottom) by the TCNN errors of Fig. 6. Evidently, poor performing clusters typically contain non-frontal faces, with the lowest performance in Fig. 1 (bottom) on near-profile views. These poses are underrepresented in the training set and so it is no surprise that detection accuracy is lower in those clusters. A different reason for higher errors on non-frontal images relates to the error measure: The detection error rate [34] is normalized by the inter-ocular distance which is smaller for profile faces, even if the face size remains the same, thereby inflating errors for those images. Importantly, regardless of the reason, images in these non-frontal clusters are where TCNN was the most effective, providing the largest improvements in accuracy. The same improvement in non-frontal faces is evident when comparing our TCNN with the direct regression method, SDM [5] (in pink).

Benchmarks. We evaluate our TCNN on two standard benchmarks for five-point landmark detection: the Annotated Face in-the-Wild (AFW) [2], containing 468 faces, and the Annotated Facial Landmarks in the Wild (AFLW) [33] with its 24,386 faces, using the same test subsets from Zhang et al. [16]. Both sets include faces from Flickr albums, manually annotated with five facial landmarks. Both therefore represent unconstrained settings with many of the appearance variations our method is expected to handle.

Effect of K. We evaluate how the number of clusters, K, affects overall accuracy. Fig. 7 reports error rates for both the AFW and AFLW benchmarks with varying cluster numbers. Using more tweaked final layers appears to improve performance. This, however, is true only up to a point; splitting the training data into too many clusters produces clusters which do not have enough examples for effective fine-tuning. In fact, beyond $K = 64$ clusters, fine-tuning the final network layers often resulted in overfitting. The results reported next use 48 clusters for our TCNN implementation.

Also interesting is the tweaking effort (not shown). Measured in epoch numbers, the effort ($\pm$ SD) required to fine-tune each tweaked process for the different values of K is 79.63 ± 2.9. Hence, this effort grows linearly with K.

Comparison with state of the art. The following facial landmark detectors are compared with our approach: ESR [4], CDM [9], RCPR [10], SDM [5], TCDCN'14 [16][2] and BB-FCN [13]. Fig. 8 (left) shows our results vs. those reported by Liang et al. [13] on the same benchmark protocols. The only exception, BB-FCN, was trained on twice the images. Fig. 8 (right) also provides accumulative error curves for methods which reported this information. Lastly, Fig. 9 illustrates some TCNN detections.

Our vanilla CNN is on par with the state of the art. TCNN improves this, surpassing previous results in both benchmarks, cutting **8%** of the error on AFLW and **17%** on AFW. To emphasize the significance of this, we note that performance gaps between us and the best published result [16] are 1.4% (AFW) and 0.62% (AFLW), whereas their reported gaps were 0.6% and 0.5%, respectively.

Furthermore, the combined benefit of more data and larger networks was previously noted by, e.g., Simonyan and Zisserman [35], and can at least partially explain the performance of BB-FCN [13]. Our results show that *better performance* can be achieved by careful processing using a *simpler model* and *less data*.

5.2 49 / 68 landmark detection

We test the accuracy obtained when using our system to detect 49 and 68 facial landmarks. Our system is not specifically tailored to particular landmark numbers or arrangements. Direct detection of 49 or 68 landmarks, however, implies a much higher dimensional regression problem (e.g., 49 landmarks translate to a 98D regression problem). This requires substantially more training data than was available in standard training sets, even with our alignment-sensitive data augmentation (Section 3.2). Rather than retraining our system for these tasks, we therefore instead simply use its five landmarks to initialize an existing, publicly available 49- or 68-point landmark detector. In our tests, we used two such methods for this purpose: CLNF [39] and the more recent DCLM [40].

We evaluate performance on the 300W data set [41], the most challenging benchmark of its kind. As a training set for these experiments, we used landmark annotated images from the HELEN training set [42] (2000 images) and the LFPW training set [43] (811). This provided a total of 2811 training images. As a test set, we used the LFPW test set

2. Different results were reported for TCDN in the original conference paper from 2014 [16] and its 2015 journal version [17]. We compare with both, and, when relevant, denote the year.

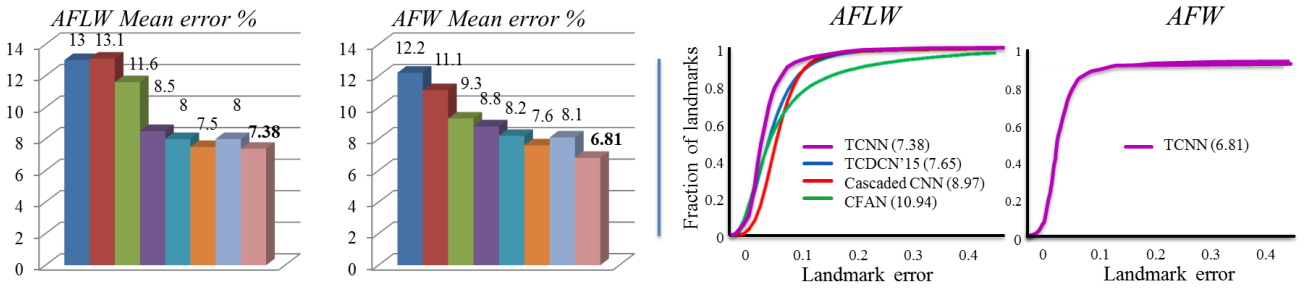
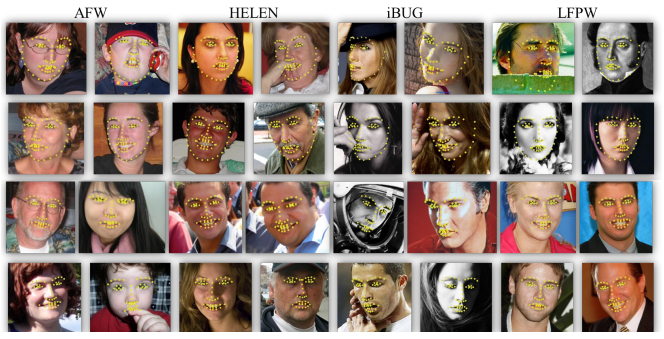


Fig. 8. *AFLW and AFW results.* Left: Error rates on AFLW [33] and AFW [2] for ESR [4], CDM [9], RCPR [10], SDM [5], TCDCN'14 [16] and BB-FCN [13], vs. our vanilla CNN (Sec. 2.1) and TNCC. Lower values are better. Right: Accumulative error curves reported on AFLW for CFAN [15], Cascaded CNN [14] and TCDCN'15 [17] vs. our TCNN (mean error % in parentheses). Previous methods did not report these curves for AFW.

Fig. 9. *Qualitative detections.* AFLW [33] (left) and AFW [2] (right). Ground truth landmark in white; TCNN in cyan. Typical errors highlighted in red.



(a) Qualitative detections

Method	49 Points @5%	@10%	AUC	MER	68 Points @5%	@10%	AUC	MER
Tree-based [2][1]	37.52	91.52	.683	.0626	11.6	78.75	.587	.0828
DRMF [36][1]	54.00	91.91	.713	.0564	36.84	87.04	.661	.0675
GN-DPM [37]	64.62	72.61	.638	.1141	-	-	-	-
SDM [5][1]	74.27	83.14	.682	.0411	-	-	-	-
ERT [38]	70.37	80.12	.706	.0892	66.76	79.63	.683	.0894
CFSS [26][2]	75.18	96.23	.777	.0460	52.83	92.74	.714	.0586
CLNF [39]	76.51	94.83	.781	.0483	52.63	89.86	.709	.0630
DCLM [40]	80.60	95.91	.793	.0460	66.37	94.64	.746	.0555
VanillaCNN+CLNF	78.39	98.26	.786	.0433	54.23	96.51	.735	.0537
TweakedCNN+CLNF	79.14	98.64	.788	.0429	54.48	97.17	.738	.0531
VanillaCNN+DCLM	86.94	98.05	.817	.0373	71.73	97.17	.767	.0473
TweakedCNN+DCLM	**88.30**	**98.83**	**.817**	**.0371**	**72.12**	**98.05**	**.771**	**.0465**

(b) Quantitative results

Fig. 10. *300W results.* (a) TCNN + CLNF qualitative results for 49 (top two rows) and 68 landmarks (bottom). Column pairs show images from the 300W subsets: AFW, HELEN, iBUG and LFPW; (b) Quantitative results: % images with 49 (68) landmark detection errors lower than 5% (10%) inter-ocular distances, area under the error curves (AUC), and mean error rates (MER). Best scores in boldface.[1] MER only for true positive detected faces. [2] Does not include AFW.

(224 images), the HELEN test set (330), AFW [2] (337), and IBUG [44] (135). In total 1026 images were used. These, collectively, form the 300W test set. Note that unlike others, we did not use AFW to train our method, allowing us to use it for testing. Once again, we use the DLIB face detector [20] to find face bounding boxes for our method. Whenever it failed to detect a face, we defaulted to the ground truth bounding boxes provided in 300W and used by the other baselines. The five points detected by our method are then used to initialize CLNF [39] and DCLM [40]. Some qualitative results of our method are provided in Fig. 10 (a).

We compare our results to the tree-based method [2], DRMF [36], GN-DPM [37], ERT [38], SDM [5], CFSS [26], CLNF [39], and DCLM [40] with their original initializations. Note that in previous reports for the performance of the tree-based method, DRMF, and SDM reflect accuracy only on true-positive detected faces (generally considered easier). Moreover, CFSS was not tested on the AFW subset of 300W. We report previously published results [39] for Tree-based, DRMF, and SDM. GN-DPM and SDM do not provide 68 point detections. Fig. 10 (b) reports these numbers. Fig. 11 additionally offers accum. errors curves for 49 and 68 point detections. Evidently, the better initialization offered by our method allows accurately localizing landmarks for a larger number of faces. To our knowledge, combining our TCNN with DCLM obtains state of the art accuracy on this benchmark.

6 Conclusions

The pursuit of better landmark detection accuracy has led many to propose progressively more elaborate models and representations, and to use increasing amounts of data to train them. Contrary to this, we show how hierarchical, discriminative processing can naturally be introduced, in an unsupervised manner, to an existing CNN design for facial landmark regression by using careful analysis and processing of the values produced at intermediate CNN layers. By so doing, we boost performance beyond that of more involved, state-of-the-art systems.

Acknowledgments

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA 2014-14071600010. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ODNI,

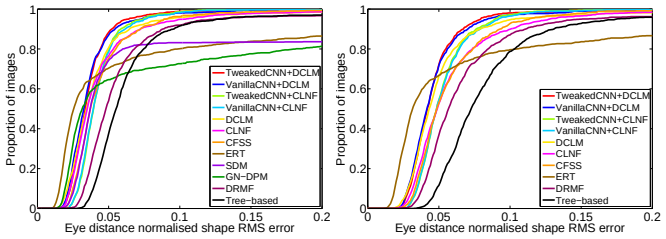


Fig. 11. *300W Accumulative error curves.* (a) 49 and (b) 68 points.

REFERENCES

[1] M. Demirkus, D. Precup, J. J. Clark, and T. Arbel, "Hierarchical temporal graphical model for head pose estimation and subsequent attribute classification in real-world videos," *Comput. Vision Image Understanding*, vol. 136, pp. 128–145, 2015.

[2] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Proc. Conf. Comput. Vision Pattern Recognition*. IEEE, 2012, pp. 2879–2886.

[3] X. Ding, W.-S. Chu, F. De la Torre, J. F. Cohn, and Q. Wang, "Facial action unit event detection by cascade of tasks," in *Proc. Int. Conf. Comput. Vision*. IEEE, 2013.

[4] X. Cao, Y. Wei, F. Wen, and J. Sun, "Face alignment by explicit shape regression," *Int. J. Comput. Vision*, vol. 107, no. 2, 2014.

[5] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in *Proc. Conf. Comput. Vision Pattern Recognition*. IEEE, 2013.

[6] T. Hassner, S. Harel, E. Paz, and R. Enbar, "Effective face frontalization in unconstrained images," in *Proc. Conf. Comput. Vision Pattern Recognition*, 2015.

[7] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. Conf. Comput. Vision Pattern Recognition*. IEEE, 2014.

[8] T. Baltrusaitis, P. Robinson, and L.-P. Morency, "Constrained local neural fields for robust facial landmark detection in the wild," in *Proc. Conf. Comput. Vision Pattern Recognition Workshops*. IEEE, 2013, pp. 354–361.

[9] X. Yu, J. Huang, S. Zhang, W. Yan, and D. N. Metaxas, "Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model," in *Proc. Int. Conf. Comput. Vision*. IEEE, 2013, pp. 1944–1951.

[10] X. P. Burgos-Artizzu, P. Perona, and P. Dollár, "Robust face landmark estimation under occlusion," in *Proc. Int. Conf. Comput. Vision*. IEEE, 2013, pp. 1513–1520.

[11] G. Ghiasi and C. C. Fowlkes, "Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model," in *Proc. Conf. Comput. Vision Pattern Recognition*. IEEE, 2014.

[12] H. Yang, W. Mou, Y. Zhang, I. Patras, H. Gunes, and P. Robinson, "Face alignment assisted by head pose estimation," in *Proc. British Mach. Vision Conf.*, 2015.

[13] Z. Liang, S. Ding, and L. Lin, "Unconstrained facial landmark localization with backbone-branches fully-convolutional networks," *arXiv preprint arXiv:1507.03409*, 2015.

[14] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *Proc. Conf. Comput. Vision Pattern Recognition*. IEEE, 2013.

[15] J. Zhang, S. Shan, M. Kan, and X. Chen, "Coarse-to-fine auto-encoder networks (CFAN) for real-time face alignment," in *European Conf. Comput. Vision*. Springer, 2014.

[16] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *European Conf. Comput. Vision*. Springer, 2014, pp. 94–108.

[17] ——, "Learning deep representation for face alignment with auxiliary attributes," *Trans. Pattern Anal. Mach. Intell.*, 2015, to appear.

[18] M. Aubry and B. C. Russell, "Understanding deep features with computer-generated imagery," in *Proc. Int. Conf. Comput. Vision*, 2015.

[19] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Neural Inform. Process. Syst.*, 2014, pp. 3320–3328.

[20] D. E. King, "Dlib-ml: A machine learning toolkit," *J. Mach. Learning Research*, vol. 10, pp. 1755–1758, 2009.

[21] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Int. Conf. on Learning Representations*, 2014.

[22] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," UMass, Amherst, Tech. Rep. 07-49, October 2007.

[23] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[24] C. Cao, Q. Hou, and K. Zhou, "Displaced dynamic expression regression for real-time facial tracking and animation," *ACM Trans. on Graphics*, vol. 33, no. 4, p. 43, 2014.

[25] G. Levi and T. Hassner, "Age and gender classification using convolutional neural networks," in *Proc. Conf. Comput. Vision Pattern Recognition Workshops*, June 2015. [Online]. Available: \url{http://www.openu.ac.il/home/hassner/projects/cnn_agegender}

[26] S. Zhu, C. Li, C. Change Loy, and X. Tang, "Face alignment by coarse-to-fine shape searching," in *Proc. Conf. Comput. Vision Pattern Recognition*, 2015, pp. 4998–5006.

[27] H. Yang and I. Patras, "Mirror, mirror on the wall, tell me, is the error small?" in *Proc. Conf. Comput. Vision Pattern Recognition*, 2015.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Neural Inform. Process. Syst.*, 2012, pp. 1097–1105.

[29] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[30] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. Conf. Comput. Vision Pattern Recognition*. IEEE, 2014, pp. 1717–1724.

[31] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proc. Int. Conf. Comput. Vision*, 2015.

[32] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. Int. Conf. Comput. Vision*, 2015.

[33] M. Köstinger, 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 *Proc. Int. Conf. Comput. Vision Workshops*. IEEE, 2011.

[34] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool, "Real-time facial feature detection using conditional regression forests," in *Proc. Conf. Comput. Vision Pattern Recognition*. IEEE, 2012.

[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Int. Conf. on Learning Representations*, 2015.

[36] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic, "Robust discriminative response map fitting with constrained local models," in *Proc. Conf. Comput. Vision Pattern Recognition*, 2013.

[37] G. Tzimiropoulos and M. Pantic, "Gauss-newton deformable part models for face alignment in-the-wild," in *Proc. Conf. Comput. Vision Pattern Recognition*, 2014, pp. 1851–1858.

[38] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proc. Conf. Comput. Vision Pattern Recognition*. IEEE, 2014.

[39] T. Baltrušaitis, P. Robinson, and L.-P. Morency, "Continuous conditional neural fields for structured regression," in *European Conf. Comput. Vision*. Springer, 2014, pp. 593–608.

[40] A. Zadeh, T. Baltrušaitis, and L.-P. Morency, "Deep constrained local models for facial landmark detection," *arXiv preprint arXiv:1611.08657*, 2016.

[41] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: Database and results," *Image and Vision Computing*, 2015.

[42] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang, "Interactive facial feature localization," in *European Conf. Comput. Vision*. Springer, 2012, pp. 679–692.

[43] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar, "Localizing parts of faces using a consensus of exemplars," *Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2930–2940, 2013.

[44] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: The first facial landmark localization challenge," in *Proc. Conf. Comput. Vision Pattern Recognition Workshops*, 2013.