DeCaFA: Deep Convolutional Cascade for Face Alignment In The Wild

Arnaud Dapogny[1,2], Kevin Bailly[2, 3], and Matthieu Cord[1]

[1]LIP6, Sorbonne Université, CNRS, 4 place Jussieu, 75005 Paris
[2]Datakalab, 114 boulevard Malesherbes, 75017 Paris
[3]ISIR, Sorbonne Université, CNRS, 4 place Jussieu, 75005 Paris

Abstract

Face Alignment is an active computer vision domain, that consists in localizing a number of facial landmarks that vary across datasets. State-of-the-art face alignment methods either consist in end-to-end regression, or in refining the shape in a cascaded manner, starting from an initial guess. In this paper, we introduce DeCaFA, an end-to-end deep convolutional cascade architecture for face alignment. DeCaFA uses fully-convolutional stages to keep full spatial resolution throughout the cascade. Between each cascade stage, DeCaFA uses multiple chained transfer layers with spatial softmax to produce landmark-wise attention maps for each of several landmark alignment tasks. Weighted intermediate supervision, as well as efficient feature fusion between the stages allow to learn to progressively refine the attention maps in an end-to-end manner. We show experimentally that DeCaFA significantly outperforms existing approaches on 300W, CelebA and WFLW databases. In addition, we show that DeCaFA can learn fine alignment with reasonable accuracy from very few images using coarsely annotated data.

1. Introduction

Face alignment consists in localizing landmarks (e.g. lips and eyes corners, pupils, nose tip). It is an important computer vision field, as it is essential for expression analysis [28], face recognition [19], tracking [2], and synthesis [20].

Recent face alignment approaches either belongs to cascaded regression or deep end-to-end regression methods. On the one's hand, cascaded regression consists in learning a sequence of updates, starting from an initial guess, to refine the landmark localization in a coarse-to-fine manner. This allows to robustly learn rigid transformations, such as translation and rotation, in the first cascade stages, then learning non-rigid deformation (e.g. due to facial expression).

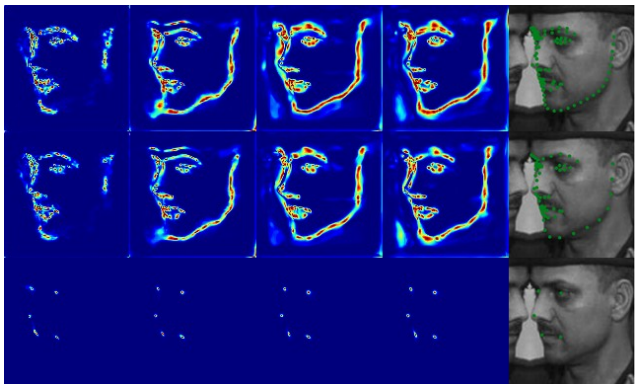

Figure 1. DeCaFA estimates landmark-wise attention maps at several stages of its architecture (horizontally: stages 1 to 4). By chaining transfer layers, it can integrate heterogeneous data (Vertically: attention maps and predictions for 98, 68 and 5-landmarks.

On the other hand, many deep approaches aim at aligning the landmarks from the image directly. However, because annotating landmarks is tedious, data is scarce and the nature of the annotations vary a lot. Thus, end-to-end approaches usually rely on learning intermediate representation (e.g. edges) to drive the alignment process. However, these representations are *ad hoc* and sub-optimal for localizing landmarks.

In this paper, we introduce a Deep convolutional Cascade for Face Alignment (DeCaFA). It contains several stages producing attention maps, relatively to heterogeneous landmark annotation markups. Figure 1 shows attention maps extracted by the subsequent stages (horizontally) and for three markups (vertically). These attention maps are refined through the successive stages for each of these markups. The contributions of this paper are thus three-fold:

- We introduce a fully-convolutional Deep Cascade for Face Alignment (DeCaFA) that unifies cascaded regression and end-to-end deep approaches, by using landmark-wise attention maps fused to extract local information around a current landmark estimate.

- We show that intermediate supervision with increasing weights helps DeCaFA learn coarse attention maps in its early stages, that are refined later on. Through chaining multiple transfer layers, DeCaFA integrates heterogeneous data and model inter-task relationships.

- We show experimentally that DeCaFA significantly outperforms existing approaches on multiple datasets, inluding the recent WFLW database. Additionally, we highlight how coarsely annotated data helps learn fine landmark alignment even with few annotated images.

2. Related work

Popular examples of cascaded regression methods include SDM [25]. In their pioneering work, Xiong *et al.* show that using simple linear regressors upon SIFT features in a cascaded manner provides precise alignment. LBF [16] is a refinement that employs randomized decision trees to dramatically speed up feature extraction. DAN [8] uses deep networks to learn each cascade stage. However, one downside of these approaches is that the update regressors are not learned jointly in a end-to-end fashion, thus there is no guarantee that the learned feature point alignment sequences is optimal. MDM [21] improves the feature extraction process by sharing the convolutional layer among all steps of the cascade that are performed through a recurrent neural network. This results in memory footprint reduction as well as a more optimized landmark trajectory throughout the cascade.

TCDCN [29] was perhaps the first end-to-end framework that could compete with cascaded regression approaches. It relies on supervised pretraining on a wide database of facial attributes. More recently, PCD-CNN [9] uses head pose to drive training. CPM+SBR [5] employs landmark registration to regularize training. SAN [4] uses adversarial networks to convert images from different styles to an aggregated style, upon which regression is performed. In [22] the authors propose to use edge map estimation as an intermediate representation to drive the landmark prediction task. Finally, DSRN [15] relies on Fourier embedding and low-rank learning to produce such representation. However, the use of such representation is usually ad hoc and it is hard to know which one would be all-around better for face alignment. Recently, AAN [26] proposes to use intermediate feature maps as attentional masks to select relevant regions. It also uses intermediate supervision to constrain those maps to correspond to landmark-wise attention maps. However, there is no guarantee that the network will learn to align landmarks in a cascaded, coarse-to-fine manner.

Furthermore, annotating images in term of several face landmarks is a time-consuming task. As a result, data is rather scarce and annotated in terms of varying number of landmarks. For instance, 300W database [17] contains approximately 3000 images labelled with 68 landmarks for train, whereas WFLW database [22] contains 7500 images with 98 landmarks. Thus, one can wonder if we can use all those images within the same framework to learn more robust landmark predictions, and if coarsely annotated data (e.g. in terms of 5 landmarks [11]) would be of any help to address finer tasks. In [23] the authors address this problem by using a classical multi-task formulation. However, this essentially ignores the intrinsic relationship between the structure of different landmark alignment tasks. Likewise, if we can predict the position of 68 landmarks, we can also easily deduce the position of landmarks for a coarser markup, such as eye/mouth corners and nose tip [11]. Authors of [27] propose to predict the union of all landmarks, with a sparse shape regression pipeline for inferring the missing landmarks for one markup. However, this method requires the numbers of landmarks to be roughly equivalent since a fine-grained (e.g. 98 landmarks) can hardly be converted into a very coarse markup (e.g. 5 points). DeFA [10] proposes to unify all the sparse landmark alignment task into a dense model fitting, however such models usually struggle with large face deformations, e.g. due to facial expressions.

3. DeCaFA overview

In this Section, we introduce our Deep convolutional Cascade for Face Alignment (DeCaFA), as illustrated on Figure 2. DeCaFA consists of S stages, each of which contains a fully-convolutional U-net backbone that preserves the full spatial resolution, as well as an attention map generation subnetwork. Section 3.1 shows how we derive landmark-wise attention maps for one landmark prediction task. Section 3.2 explains how several transfer layers can be chained to produce such attention maps, relatively to K landmark prediction tasks. The input of the next stage is obtained by applying a fusion algorithm that involves the attention maps, as explained in Section 3.3. In Section 3.4 we describe how DeCaFA is trained in an end-to-end manner with weighted intermediate supervision. Finally, in Section 3.5 we provide implementation details to facilitate reproducibility.

3.1. Landmark-wise attention maps

The U-net at stage i takes an input I_i and gives rise to an embedding H_i and parameters θ_i. In order to produce a suitable embedding from H_i for predicting L landmarks, we apply a 1×1 convolutional layer with L filters with parameters θ_i'. We denote the embeddings outputted by this transfer layer as T_i^L. In order to highlight its dominant mode we apply a spatial softmax operator. Formally, for a pixel with coordinates (x, y) and a landmark l:

$$\Phi_i^L(x, y, l) = \frac{\exp(T_i^L(x, y, l))}{\sum_{x=1}^{X} \sum_{y=1}^{Y} \exp(T_i^L(x, y, l))} \quad (1)$$

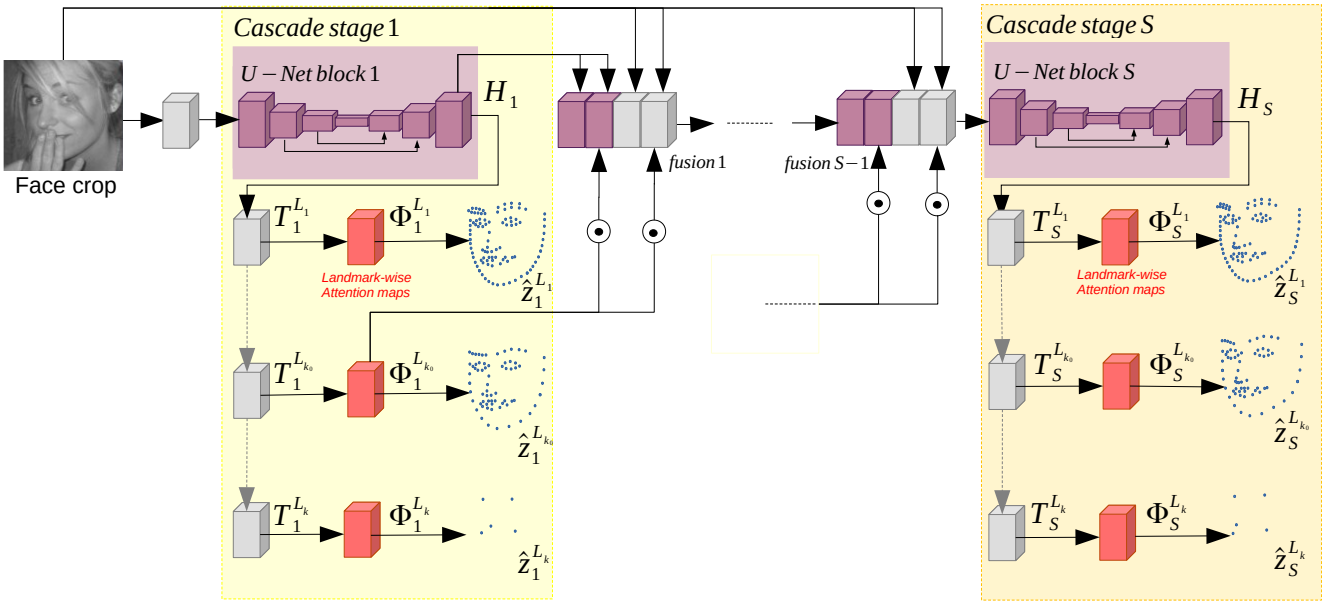

Figure 2. DeCaFA architecture overview. Several stages with fully-convolutional U-nets are stacked, multiple transfer layers are chained and intermediate supervision with increasing weights is applied to produce landmark estimates for heterogeneous alignment tasks. Landmark-wise attention maps are fused with the input image and the embeddings of the previous stage U-net to enable end-to-end cascaded alignment.

An estimation $\hat{z}_i^L$ of the landmark coordinates can be obtained by computing the first order moments of Φ_i^L:

$$\begin{cases} \hat{z}_{i,x}^L(l) = \mathbb{E}_{x,y}[x\Phi_i^L(x,y,l)] \\ \hat{z}_{i,y}^L(l) = \mathbb{E}_{x,y}[y\Phi_i^L(x,y,l)] \end{cases} \quad (2)$$

Where $\hat{z}_{i,x}^L$ and $\hat{z}_{i,y}^L$ are two vectors of size L containing the x and y landmark coordinates $\hat{z}_i^L$. The soft-argmax operator is inspired by the work in [13] in the frame of human pose estimation and provides differentiable landmark coordinates estimate from the attention map Φ_i^L.

3.2. Chaining landmark localization tasks

As it will be explained in Section 4.1, existing datasets for face alignment usually have heterogeneous annotations and varying numbers of annotated landmarks. In order to deal with these heterogeneous annotations, we integrate K tasks that consist in predicting various numbers of landmarks $L_1, ...L_K$ with $\forall k_1, k_2, k_1 \leq k_2 \implies L_{k_1} > L_{k_2}$ (i.e. we chain the landmark-wise attention maps in an decreasing order of the number of landmarks to predict). To do so, we apply K transfer layers $T_i^{L_1}, ..., T_i^{L_K}$ with parameters $\theta_i^{(1)}, ..., \theta_i^{(K)}$, at it is depicted on Figure 3 (a). We have:

$$\begin{cases} \hat{z}_{i,x}^{L_k}(l) = \mathbb{E}_{x,y}[x\Phi_i^{L_k}(x,y,l)] & \forall 1 \leq k \leq K \\ \hat{z}_{i,y}^{L_k}(l) = \mathbb{E}_{x,y}[y\Phi_i^{L_k}(x,y,l)] & \forall 1 \leq k \leq K \end{cases} \quad (3)$$

The advantages of stacking the landmarks prediction pipelines in a descending order of the number of landmarks

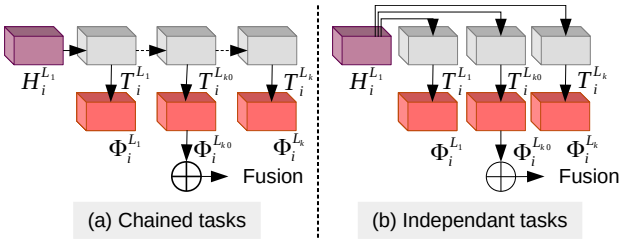

Figure 3. Chained (left) vs independant (right) task order.

to be localized are two-fold: First, from a semantic perspective, who can do more can do less, meaning that it shall be easier for the network to learn the sequence of transfer layers in that order (i.e. if we can precisely localize a 68-points markup it will be easy to also localize the nose tip, as well as mouth/eyes corners). Second, labelling images with large amounts of landmarks is a tedious task, thus generally the more annotated landmarks in a database, the less images we have at our disposal. Using such architecture ensures that the former (harder) tasks benefits from all the images annotated with the latter (easier) task. This can be seen as weakly supervised learning, where images labelled in terms of coarse markups can help to learn finer alignment tasks. Also note that as these 1×1 convolutional layers have very few parameters, thus a lot of gradient can be backpropagated down to the U-net backbone and benefit the K prediction tasks. Finally, as illustrated on Figure 3, we use attention maps $\Phi_i^{L_{k_0}}$ from markup k_0 to provide richer embeddings for the subsequent stages by applying feature fusion.

3.3. Feature fusion

In a standard feedforward deep network with S stacked stages, the $i+1^{th}$ stage takes an input $I_i = F_1$ that corresponds to the embeddings H_i outputted by the previous stage (with the convention $I_0 = I$ the original image). By contrast, in cascade-based approaches, each stage shall learn an update to bring the feature points closer to the ground truth localizations, by using information sampled around current feature point localizations. Within an end-to-end fully-convolutional deep network, an analogous statement would be that the $i+1^{th}$ stage shall use a local embedding F_2 that is calculated using information from the original image I highlighted by landmark-wise attention maps $\Phi_i^{L_{k_0}}$. In our method, we aggregate these maps by summing all the landmark-wise attention maps $M_i = \bigoplus_{l=1}^{L} \Phi_i^{L_{k_0}}$. Thus, we can write the feature fusion model for the basic deep approach as:

$$F_1(I, H_i, M_i) = H_i \tag{4}$$

and the cascade-like approach as:

$$F_2(I, H_i, M_i) = I \odot M_i \tag{5}$$

Where $\odot$ denotes the Hadamard product. This fusion scheme between the input image and the mask only preserves local information, for which the values of M_i are high. Alternatively, we can reinject the original image I inside each stage so that it can use global information in case where the mask M_i is not precise enough or contains localizations errors (as it is the case early in the cascade):

$$F_3(I, H_i, M_i) = I||(I \odot M_i) \tag{6}$$

With $||$ the channel-wise concatenation operation. Furthermore we can also fuse the relevant parts (as highlighted by mask M_i) of the embedding H_i of the previous stage U-net to provide the subsequent stages a richer, more semantically abstract information to estimate the landmarks coordinates:

$$F_4(I, H_i, M_i) = I||(I \odot M_i)||(H_i \odot M_i) \tag{7}$$

Finally, we can aso use global information from not only the image I, but also from the embeddings H_i:

$$F_5(I, H_i, M_i) = I||(I \odot M_i)||H_i||(H_i \odot M_i) \tag{8}$$

This fusion model is more efficient and is used in DeCaFA (Figure 2), as it allows using both global and local information around the estimated landmarks so as to learn cascade-like alignment in an end-to-end fashion.

3.4. Learning DeCaFA model

DeCaFA models can be trained end-to-end by optimizing the following loss function w.r.t. parameters of the U-nets θ_i and $\theta_i^{(1)}, ..., \theta_i^{(K)}$ for the transfer layers $T_i^{L_1}, ..., T_i^{L_K}$ respectively, $\forall 1 \leq k \leq K$:

$$\mathcal{L}(\theta_1, \theta_1^{(1)}, ..., \theta_1^{(K)}, ..., \theta_S, \theta_S^{(1)}, ..., \theta_S^{(K)}) = \sum_{k=1}^{K} \frac{1}{L_k} |\hat{z}_S^{L_k} - z^{L_k *}| \tag{9}$$

With $z^{L_k *}$ the ground truth landmark position for a L_k-landmarks markup. In practice, the summation in equation (9) have less terms since usually each example is annotated with only one markup. With this configuration, however, if the whole network is deep enough, few gradient will ever pass through the firsts attention maps. Even worse, there is no guarantee that these feature maps will correspond to landmark-wise attention maps in the early stages, which is key to ensure cascade-like behavior of DeCaFA. To ensure this, we add a differentiable soft-argmax layer after each spatial softmax and a supervised cost at stage i:

$$\mathcal{L}(\theta_1, \theta_1^{(1)}, ..., \theta_1^{(K)}, ..., \theta_S, \theta_S^{(1)}, ..., \theta_S^{(K)}) = \sum_{i=1}^{S} \lambda_i \sum_{k=1}^{K} \frac{1}{L_k} |\hat{z}_i^{L_k} - z^{L_k *}| \tag{10}$$

In practice, we use a $\mathcal{L}_1$ loss function, as it has been shown to overfit less on very bad examples and lead to more precise results for face alignment. However, we need to make sure that the (relatively) shallow sub-networks does not overfit on these losses, which would result in very narrow heat maps with very localized dominant modes early in the cascade, and thus an overall lower accuracy. This is ensured by applying increasing λ_i weights in (10).

3.5. Implementation details

The DeCaFA models that will be investigated below use 1 to 4 stages that each contains 12 3×3 convolutional layers with $64 \rightarrow 64 \rightarrow 128 \rightarrow 128 \rightarrow 256 \rightarrow 256$ channels for the downsampling portion, and vice-versa for the upsampling portion. The input images are resized to 128×128 grayscale images prior to being processed by the network. Each convolution is followed by a batch normalization layer with ReLU activation. In order to generate smooth feature maps we do not use transposed convolution but bilinear image upsampling followed with 3×3 convolutional layers. The whole architecture is trained using ADAM optimizer with a $5e^{-4}$ learning rate with momentum 0.9 and learning rate annealing with power 0.9. We apply 400000 updates with batch size 8 for each database, with alternating updates between the databases.

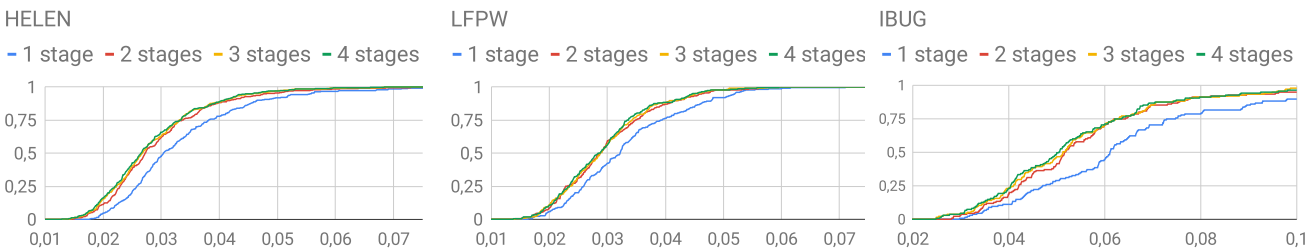

Figure 4. Comparison in terms of Cumulative error distribution (CED) curves on 300W of models with $S = 1,2,3$ and 4 stages. As we stack cascade stages, the accuracy increases and saturates after the third/fourth stage.

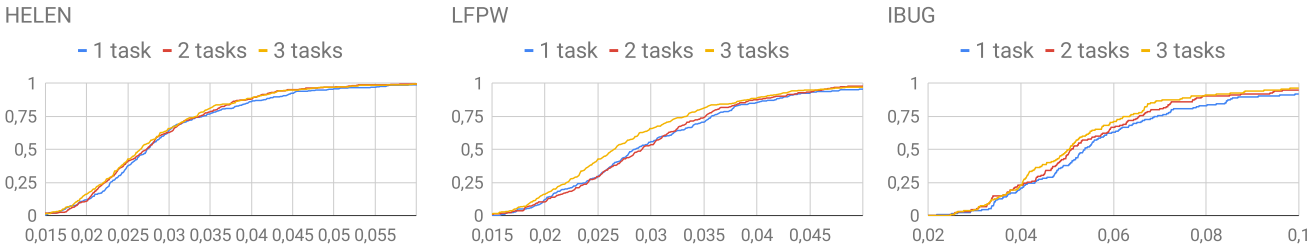

Figure 5. CED curves for models with $K = 1,2$ and 3 landmark prediction tasks. Models trained with multiple alignment tasks are significantly better.

Table 1. NME (%, (lower is better) averaged among 300W-Full, 300W-Challenging, WFLW-All, WFLW-Pose and CelebA datasets.

Fusion	task order	weights λ_i	avg. NME(%)
F_1	chained	$\uparrow$	4.83
F_2	chained	$\uparrow$	5.04
F_3	chained	$\uparrow$	4.81
F_4	chained	$\uparrow$	4.80
F_5	independant	$\downarrow$	5.11
F_5	independant	$=$	5.01
F_5	independant	$\uparrow$	4.75
F_5	chained	$\downarrow$	5.05
F_5	chained	$=$	4.91
F_5	chained	$\uparrow$	**4.69**

4. Experiments

In this section, we introduce the face alignment datasets (Section 4.1). Then, in Section 4.2 we validate hyper-parameters through ablation study. In Section 4.3 and 4.4 we compare DeCaFA with state-of-the-art approaches for alignment on still images and video, respectively. Finally, In Section 4.5 we show that DeCaFA is suitable for weakly-supervised learning with few finely-annotated examples.

4.1. Datasets

The **300W** database [17] contains moderate variations in head pose, facial expressions and illuminations. It consists in four databases: **LFPW** (811 train images / 224 test images), **HELEN** (2000 train images / 330 test images), **AFW** (337 train images) and **IBUG** (135 test images), for a total of 3148 images annotated with 68 landmarks for training the models. For comparison with state-of-the art methods, we refer to LFPW and HELEN test sets as the *common* subset and I-BUG as the *challenging* subset of 300W.

CelebA [11] is a large-scale face attribute database containing $202k$ images from $10k$ identities, each annotated with 5 landmarks (nose, left and right pupils, mouth corners). In our experiments, we train our models using the train partition that contains $16k$ images from $8k$ ids. The test set contains $20k$ instances from $1k$ ids.

The **Wider Facial Landmarks in the Wild or WFLW** database [22] contains 10000 faces (7500 for training and 2500 for testing) with 98 annotated landmarks. This database also features rich attribute annotations in terms of occlusion, head pose, make-up, illumination, blur and expressions.

The **300VW** database [18] is a video alignment database containing 114 videos making a total of 218,595 frames, which are divided into three subsets of various difficulty (categories A, B and C, C being the most challenging).

In what follows, and unless stated otherwise, we train our models using a concatenation of the train partitions of 300W, WFLW and CelebA, and evaluate on the test partition of these datasets. As in [25, 16, 30, 29, 15, 14, 24, 7] we measure the average point-to-point euclidean distance between feature points (NME), normalized by the inter-ocular distance (distance between outer eye corners). We also report AUC and failure rates for a maximum error of 0.1, as well as cumulative error distribution (CED) curves.

4.2. Ablation study

In this section, we validate the architecture and hyperparameters of our model: the number of stages S, the number of landmark prediction tasks K, the fusion and task ordering scheme as well as the intermediate supervision weights. Figure 4 shows CED curves for models with $S = 1, 2, 3$ and 4 cascade stages. The accuracy steadily increases as we add more stages, and saturates after the third on LFPW and HELEN, which is a well-known behavior of cascaded models [25, 16], showing that DeCaFA with weighted intermediate supervision indeed works as a cascade, by first providing coarse estimates and refining in the later stages. On IBUG, this difference is more conspicuous, thus there is room for improvement by stacking more cascade stages.

Figure 5 shows the interest of chaining multiple tasks, most notably on LFPW, that contains low-resolution images, and IBUG, which contains strong head pose variations as well as occlusions. Coarsely annotated data (5 landmarks) significantly helps the fine-grained landmark localization, as it is integrated a kind of weakly supervised scheme. This will be discussed more thoroughly in Section 4.5.

Table 1 shows a comparison between multiple fusion, task ordering and intermediate supervision weighting schemes. We test our model on 300W (full and challenging), WFLW (All and challenging, *i.e.* pose subset) as well as CelebA and report the average accuracy on those 5 subsets. First, reinjecting the whole input image (F_3 - Eq. (6) vs F_2 - Eq. (5)) significantly improves the accuracy, most notably on challenging data such as 300W-challenging or WFLW-pose, where the first cascade stages may commit errors. F_4 - Eq. (7) and F_3 fusion (cascaded models) using local+global information rivals the basic deep approach F_1 - Eq. (4). Furthermore, F_5 - Eq. (8) fusion, which uses local and global cues is the best by a wide margin.

Furthermore, chaining the transfer layers (Figure 3-a) is better than using independant transfer layers (Figure 3-b): likewise, in such a case, the first transfer layer benefits from the gradients from the subsequents layer at train time. Last but not least, using increasing intermediate supervision weights in Equation (10) (*i.e.* $\lambda_1 = 1/8$, $\lambda_2 = 1/4$, $\lambda_3 = 1/2$, $\lambda_4 = 1$) is better than both using constant weights ($\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1$) and decreasing weights ($\lambda_1 = 1$, $\lambda_2 = 1/2$, $\lambda_3 = 1/4$, $\lambda_4 = 1/8$), as it enables proper cascade-like training, the first stages of the network outputting coarse attention maps that are refined later on.

4.3. Comparisons with state-of-the-art methods

Table 3 shows a comparison between DeCaFA and recent state-of-the-art approaches on 300W database. Our approach performs better than most existing approaches on the common subset, and performs very close to its best contenders on the challenging subset. Note that DeCaFA trained only on 300W trainset has a NME of 3.69% and is already very competitive with recent approaches [9, 5, 4, 8], thanks to its end-to-end cascade architecture. DeCaFA is competitive with the best approaches, LAB [22] and DAN-MENPO [8] as well as JMFA-MENPO [3], which also use external data.

Table 2 shows a comparison between our method and LAB [22] on WFLW database. As in [22] we report the average point-to-point error on WFLW test partition, normalized by the outer eye corners. We also report the error on multiple test subsets containing variations in head pose, facial expressions, illumination, make-up as well as partial occlusions and occasional blur. DeCaFA performs better than LAB [22] and Wing [6] by a significant margin on every subset. Also, note that DeCaFA trained solely on WFLW already has a NME of 5.01 on the whole test set, which is still better that these two methods. Lastly, there is room for improvement on this benchmark as we do not excplicitely handle any of the factors of variation such as pose or occlusions.

Finally, Table 5 shows a comparison of our method and state-of-the-art approaches on CelebA. As in [25, 30, 15, 26] we report the average point-to-point error on the test partition, normalized by the distance between the two eye centers. Our approach is the best by a significant margin. Noteworthy, even though we use auxiliary data from 300W and WFLW, we do not use data from the *val* partition of CelebA, contrary to [15, 26], thus there is significant room for improvement.

Overall, DeCaFA sets a new state-of-the-art on three databases with several evaluation metrics. Figure 7 provides qualitative assessment of the alignment quality, as well as visualizations of the attention maps. In addition, DeCaFA embraces few parameters ($\approx 10M$) compared to state-of-the-art approaches, and can be run at 32 fps on a GTX1060.

4.4. face alignment on video

Table 6. NME for video alignment on 300VW database.

Method	cat. A	cat. B	cat. C
DSRN [15]	5.33	4.92	8.85
SA [12]	3.85	**3.46**	7.51
DeCaFA	**3.82**	3.63	**6.67**

In this section we evaluate DeCaFA on 300VW video database. Similarly to the two-steps procedure described in [6], we train a first 10-layers CNN to correct the bounding box coordinates on WFLW. Then, for each video, we initialize the bounding box for the first frame using the ground truth landmarks. For each subsequent frame, we generate a new bounding box using the landmarks for the last frame and correct it using the bounding box correction CNN. We then align the landmarks for this frame using DeCaFA. As shown in Table 6, DeCaFA is able to outperform recent approaches on this benchmark, particularly in difficult conditions (category C) in terms of NME. It also obtains AUC/FR@0.1 **0.633/1.35**, outperforming state-of-the-art (0.594/4.57 [1]).

Table 2. Comparison in terms of NME (lower is better), AUC (higher is better) as well as failure rate (lower is better), on WFLW.

metric	method	all	head pose	expression	illumination	make-up	occlusion	blur
NME(%)	CFSS [30]	9.07	21.36	10.09	8.30	8.74	11.76	9.96
	DVLN [23]	10.84	46.93	11.15	7.31	11.65	16.30	13.71
	LAB [22]	5.27	10.24	5.51	5.23	5.15	6.79	6.32
	Wing [6]	5.11	8.75	5.36	4.93	5.41	6.37	5.81
	DeCaFA	**4.62**	**8.11**	**4.65**	**4.41**	**4.63**	**5.74**	**5.38**
AUC@0.1	CFSS [30]	0.366	0.063	0.316	0.385	0.369	0.269	0.303
	DVLN [23]	0.456	0.147	0.389	0.474	0.449	0.379	0.397
	LAB [22]	0.532	0.235	0.495	0.543	0.539	0.449	0.463
	Wing [6]	0.554	**0.310**	0.496	0.541	0.558	**0.489**	0.492
	DeCaFA	**0.563**	0.292	**0.546**	**0.579**	**0.575**	0.485	**0.494**
FR@0.1(%)	CFSS [30]	20.56	66.26	23.25	17.34	21.84	32.88	23.67
	DVLN [23]	10.84	46.93	11.15	7.31	11.65	16.30	13.71
	LAB [22]	7.56	28.83	6.37	6.73	7.77	13.72	10,74
	Wing [6]	6.00	22.70	4.78	4.30	7.77	12.50	7.76
	DeCaFA	**4.84**	**21.4**	**3.73**	**3.22**	**6.15**	**9.26**	**6.61**

Table 3. NME (%) comparison on 300W.

Method	Com.	Chall.	full
PCD-CNN [9]	3.67	7.62	4.44
CPM+SBR [5]	3.28	7.58	4.10
SAN [4]	3.34	6.60	3.98
DAN [8]	3.19	5.24	3.59
LAB [22]	2.98	5.19	3.49
DAN-MENPO [8]	3.09	**4.88**	3.44
DeCaFA	**2.93**	5.26	**3.39**

Table 4. AUC and FR(%) @0.1 on 300w.

Method	AUC	FR
CFSS [30]	49.87	5.08
Densereg+MDM [1]	52.19	3.67
JMFA [3]	54.9	1.00
JMFA-MENPO [3]	60.7	0.33
LAB [22]	58.9	0.83
DeCaFA	**66.1**	**0.15**

Table 5. NME (%) on CelebA.

Method	NME (%)
SDM [25]	4.35
CFSS [30]	3,95
DSRN [15]	3.08
AAN [26]	2.99
DeCaFA	**2.10**

4.5. Weakly supervised learning

We also study how DeCaFA learns in a weakly supervised (WSL) context using examples by using only a small fraction of 300W (100/500 images, 3% and 15% of trainset) and WFLW (100/500 images, 1% and 6% of trainset) and the whole CelebA trainset, reporting results on 300W and WFLW testsets on Figure 6. Using CelebA improves the accuracy in both cases, most notably when the number of training images is very low. For instance, DeCaFA trained with 3% of 300W and 1% of WFLW already outputs decent fine-grained landmark estimations, as it is better than CFSS [30] and DVLN ([23], see Table 2) on WFLW. DeCaFA trained with 15% of 300W and 6% of WFLW is on par with SAN on 300W ([4], see Table 3), and is better than DVLN on WFLW. This indicates that WSL involving CelebA significantly improves the accuracy for predicting 68 and 98 landmarks. Thus, due to the chaining of multiple transfer layers, DeCaFA is well suited for WSL and can be trained at a lower cost with coarsely annotated examples.

5. Conclusion

In this paper, we introduced DeCaFA for face alignment. DeCaFA unifies cascaded regression and an end-to-end trainable deep approaches by using landmark-wise attention maps to select the most relevant regions and intermediate supervisions with increasing weights to ensure proper cascaded alignment. By chaining multiple transfer layers to produce attention maps corresponding to different alignment tasks, DeCaFA benefits from heterogeneous data. We empirically show that DeCaFA significantly outperforms state-of-the-art approaches on 300W, CelebA and WFLW databases. In addition, DeCaFA is very modular and is suited for weakly supervised learning using coarsely annotated data.

Future work will consist in integrating other sources of data, or possibly other representations and tasks, such as head pose estimation, partial occlusion handling, as well as facial expressions, Action Unit and/or attributes (such as age or gender estimation) recognition within DeCaFA framework. Furthemore, we will study the application of DeCaFA to closely related fields, such as human pose estimation.

Acknowledgments

This work was partially supported by the French National Agency (ANR) in the frame of its Technological Research (DS0705) 2016 program (Deep in France, project number ANR-13-CORD-0004) and its Technological Research JCJC program (FacIL, project ANR-17-CE33-0002).

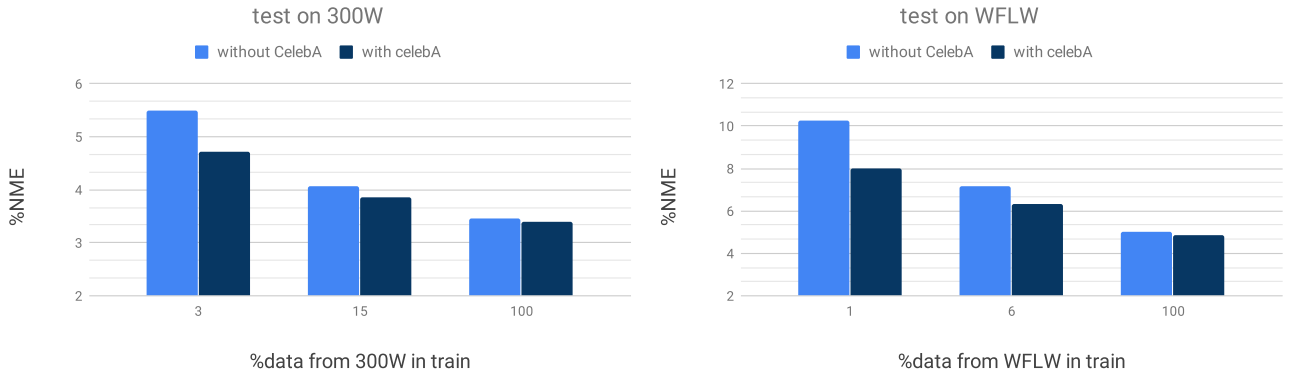

Figure 6. % mean error comparison when training with small fraction of the training set and coarsely annotated examples from CelebA.

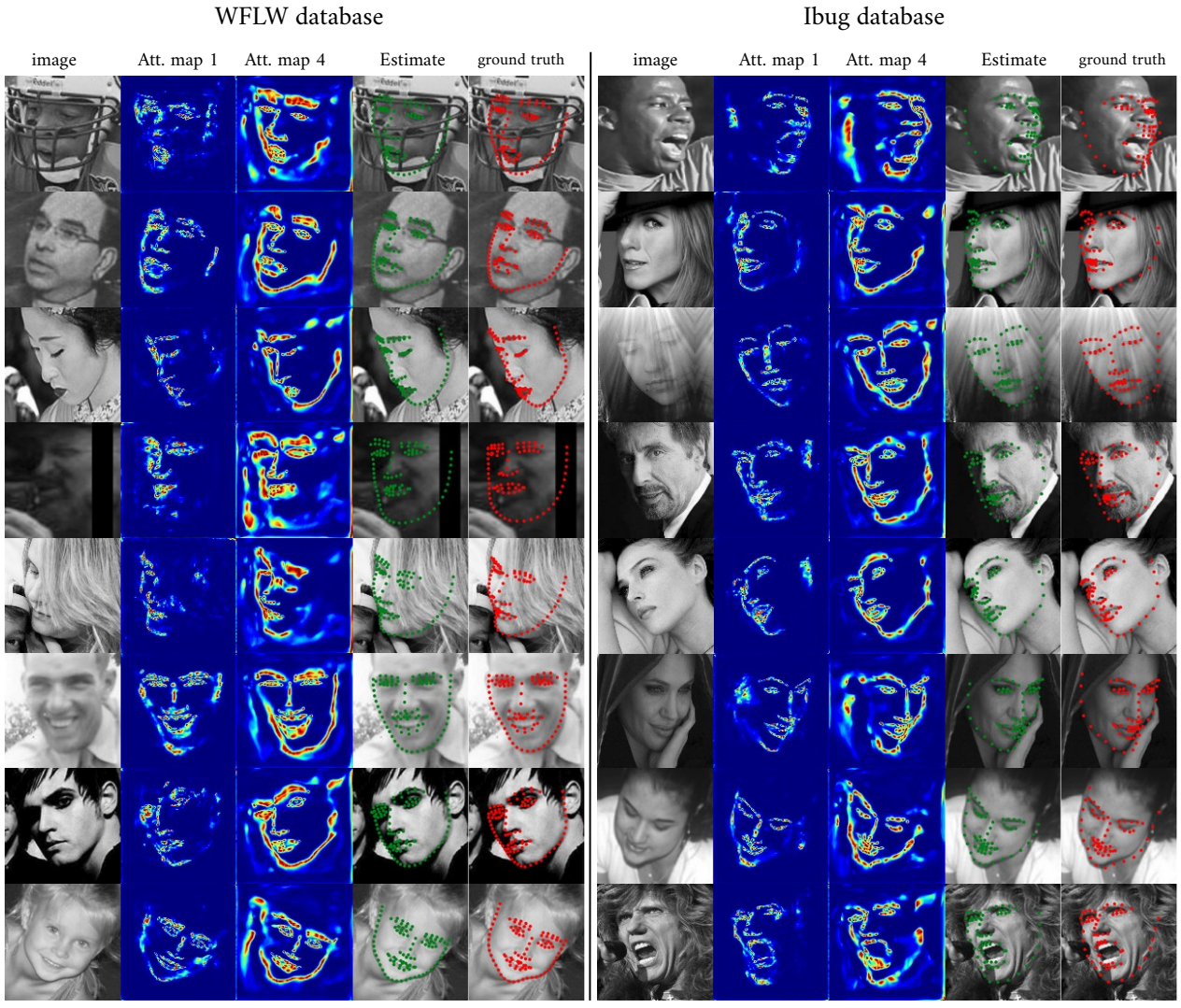

Figure 7. From left to right: images, attention maps outputted by stages 1 and 4, alignment results, and ground truth for images from 300W (I-bug, 68 landmarks) and WFLW (98 landmarks). Notice how the summed attention maps are iteratively refined, and how closely the predicted landmarks usually matches the ground truth, even under difficult illumination, non-frontal head poses, make-up, or occlusions.

References

[1] Riza Alp Guler, George Trigeorgis, Epameinondas Antonakos, Patrick Snape, Stefanos Zafeiriou, and Iasonas Kokkinos. Densereg: Fully convolutional dense shape regression in-the-wild. In *CVPR*, 2017. 6, 7

[2] Grigorios G Chrysos, Epameinondas Antonakos, Patrick Snape, Akshay Asthana, and Stefanos Zafeiriou. A comprehensive performance evaluation of deformable face tracking "in-the-wild". *IJCV*, 2018. 1

[3] Jiankang Deng, George Trigeorgis, Yuxiang Zhou, and Stefanos Zafeiriou. Joint multi-view face alignment in the wild. *arXiv preprint arXiv:1708.06023*, 2017. 6, 7

[4] Xuanyi Dong, Yan Yan, Wanli Ouyang, and Yi Yang. Style aggregated network for facial landmark detection. In *CVPR*, 2018. 2, 6, 7

[5] Xuanyi Dong, Shoou-I Yu, Xinshuo Weng, Shih-En Wei, Yi Yang, and Yaser Sheikh. Supervision-by-registration: An unsupervised approach to improve the precision of facial landmark detectors. In *CVPR*, 2018. 2, 6, 7

[6] Zhen-Hua Feng, Josef Kittler, Muhammad Awais, Patrik Huber, and Xiao-Jun Wu. Wing loss for robust facial landmark localisation with convolutional neural networks. In *CVPR*, 2018. 6, 7

[7] Sina Honari, Pavlo Molchanov, Stephen Tyree, Pascal Vincent, Christopher Pal, and Jan Kautz. Improving Landmark Localization with Semi-Supervised Learning. *CVPR*, 2018. 5

[8] Marek Kowalski, Jacek Naruniec, and Tomasz Trzcinski. Deep alignment network: A convolutional neural network for robust face alignment. In *CVPR workshops*, 2017. 2, 6, 7

[9] Amit Kumar and Rama Chellappa. Disentangling 3d pose in a dendritic cnn for unconstrained 2d face alignment. In *CVPR*, 2018. 2, 6, 7

[10] Yaojie Liu, Amin Jourabloo, William Ren, and Xiaoming Liu. Dense face alignment. In *ICCV*, 2017. 2

[11] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015. 2, 5

[12] Zhiwei Liu, Xiangyu Zhu, Guosheng Hu, Haiyun Guo, Ming Tang, Zhen Lei, Neil M Robertson, and Jinqiao Wang. Semantic alignment: Finding semantically consistent ground-truth for facial landmark detection. In *CVPR*, 2019. 6

[13] Diogo C Luvizon, David Picard, and Hedi Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *CVPR*, 2018. 3

[14] Jiang-Jing Lv, Xiaohu Shao, Junliang Xing, Cheng Cheng, Xi Zhou, et al. A deep regression architecture with two-stage re-initialization for high performance facial landmark detection. In *CVPR*, 2017. 5

[15] Xin Miao, Xiantong Zhen, Xianglong Liu, Cheng Deng, Vassilis Athitsos, and Heng Huang. Direct shape regression networks for end-to-end face alignment. In *CVPR*, 2018. 2, 5, 6, 7

[16] Shaoqing Ren, Xudong Cao, Yichen Wei, and Jian Sun. Face alignment at 3000 FPS via regressing local binary features. *CVPR*, 2014. 2, 5, 6

[17] Christos Sagonas, Epameinondas Antonakos, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 Faces In-The-Wild Challenge: database and results. *IVC*, 2015. 2, 5

[18] Jie Shen, Stefanos Zafeiriou, Grigoris G Chrysos, Jean Kossaifi, Georgios Tzimiropoulos, and Maja Pantic. The first facial landmark tracking in-the-wild challenge: Benchmark and results. In *ICCV Workshops*, 2015. 5

[19] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 2014. 1

[20] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *CVPR*, 2016. 1

[21] George Trigeorgis, Patrick Snape, Mihalis A. Nicolaou, Epameinondas Antonakos, and Stefanos Zafeiriou. Mnemonic Descent Method: A Recurrent Process Applied for End-to-End Face Alignment. *CVPR*, 2016. 2

[22] Wayne Wu, Chen Qian, Shuo Yang, Quan Wang, Yici Cai, and Qiang Zhou. Look at boundary: A boundary-aware face alignment algorithm. In *CVPR*, 2018. 2, 5, 6, 7

[23] Wenyan Wu and Shuo Yang. Leveraging intra and inter-dataset variations for robust face alignment. In *CVPR workshops*, 2017. 2, 7

[24] Shengtao Xiao, Jiashi Feng, Junliang Xing, and Hanjiang Lai. Robust Facial Landmark Detection via Recurrent Attentive-Refinement Networks. *ECCV*, 2016. 5

[25] Xuehan Xiong and Fernando De La Torre. Supervised descent method and its applications to face alignment. *CVPR*, 2013. 2, 5, 6, 7

[26] Lei Yue, Xin Miao, Pengbo Wang, Baochang Zhang, Xiantong Zhen, and Xianbin Cao. Attentional alignment network. *BMVC*, 2018. 2, 6, 7

[27] Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Leveraging datasets with varying annotations for face alignment via deep regression network. In *ICCV*, 2015. 2

[28] Yong Zhang, Rui Zhao, Weiming Dong, Bao-Gang Hu, and Qiang Ji. Bilateral ordinal relevance multi-instance regression for facial action unit intensity estimation. In *CVPR*, 2018. 1

[29] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Learning deep representation for face alignment with auxiliary attributes. *PAMI*, 2016. 2, 5

[30] Shizhan Zhu, Cheng Li, Chen Change Loy, and Xiaoou Tang. Face alignment by coarse-to-fine shape searching. In *CVPR*, 2015. 5, 6, 7