Joint Head Pose Estimation and Face Alignment Framework Using Global and Local CNN Features

Xiang Xu and Ioannis A. Kakadiaris
Computational Biomedicine Lab
Department of Computer Science, University of Houston, Houston, TX, USA
{xxu18, ikakadia}@central.uh.edu

***Abstract*— In this paper, we explore global and local features obtained from Convolutional Neural Networks (CNN) for learning to estimate head pose and localize landmarks jointly. Because there is a high correlation between head pose and landmark locations, the head pose distributions from a reference database and learned local deep patch features are used to reduce the error in the head pose estimation and face alignment tasks. First, we train *GNet* on the detected face region to obtain a rough estimate of the pose and to localize the seven primary landmarks. The most similar shape is selected for initialization from a reference shape pool constructed from the training samples according to the estimated head pose. Starting from the initial pose and shape, *LNet* is used to learn local CNN features and predict the shape and pose residuals. We demonstrate that our algorithm, named JFA, improves both the head pose estimation and face alignment. To the best of our knowledge, this is the first system that explores the use of the global and local CNN features to solve head pose estimation and landmark detection tasks jointly.**

I. INTRODUCTION

Face alignment, *a.k.a.* landmark detection, is the task of localizing the facial key points (*e.g.*, the outline of jaw, brow, nose, eyes, and mouth) on a face image. With precise landmarks, the shape and appearance of a human face can be represented easily, it serves as a necessary process for many face-related applications. In a face recognition system, whether in a 2D [1] or 3D system [12], [9], 2D landmarks can be used to transform the image to align the parts to the specific position or to estimate head pose, then fit the 3D model to the 2D face. Due to its importance, this task has been studied for many years [6], [7], [35], [5], [25], [3], [24], [13], [32], [23], [27], [22]. There has been rapid progress recently due to annotation and release of several public training datasets [20]. However, there is a critical need for improved algorithms because of the plethora of images with large variations in head pose, expression, illumination, and occlusions. Note that the expert annotators cannot maintain consistency of the landmark position. Finally, automatic face alignment systems suffer from the performance of the face detection procedure, which significantly influences the initialization of the systems.

The 300-W dataset [20] is widely used to train a landmark detection algorithm, which provides the compact bounding box generated from the annotation (Fig. 1, left column). There are two issues with using the tight bounding box in training process. First, training with this bounding box is not practical. It is hard to generalize. Second, the rectangle area annotated as bounding box contains rich and stable information for the face, which helps localize landmarks, while the bounding box by a face detector is much less robust. The differences in the bounding boxes are illustrated in Fig. 1. The bounding box generated from a face detector contains a part of the face and the background, which increases the complexity for the face alignment task.

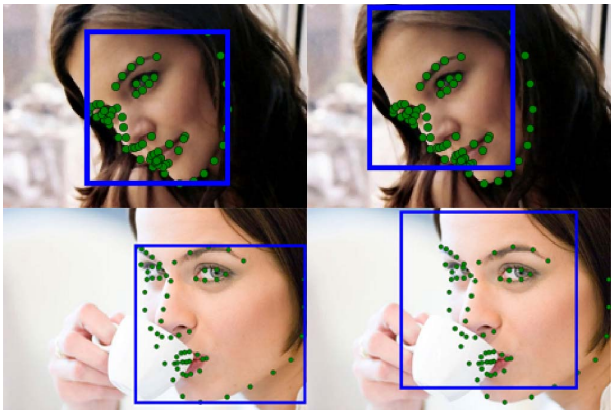


Fig. 1: Depiction of the differences in the bounding boxes. The left column demonstrates the annotated bounding box from 300-W competition. The right column is generated by the Dlib face detector. The 300-W bounding box contains the area of landmarks while the bounding box obtained by Dlib contains additional information such as hair and background.

It has been demonstrated that there is a close relationship between head pose and the distribution of the landmarks [31], [28], [33]. Using a rough head pose, the system can provide a much better initialization than the mean shape and prevent the algorithm from being stuck at a local minimum. To address the problem mentioned above and explore the benefit of using a rough estimate of head pose, we designed an efficient and robust joint head pose estimation and face alignment algorithm (JFA) with face detection results "in the wild". That is, we address the problem of face alignment with the non-ideal bounding box and solve it in a hierarchical way by exploring the relationship between head pose and landmarks using global and local features. In particular, JFA first estimates the head pose and primary points by applying a Convolutional Neural Network (CNN) named *GNet* on the

978-1-5090-4023-0/17 $31.00 © 2017 IEEE
DOI 10.1109/FG.2017.81

entire face image and initializes the shape according to the exemplars from the shape pool constructed from the training set. Then, another CNN named *LNet* is applied to learn the local features from the patches cropped from the current shape. With the local and global deep features, the head pose's residual and shape increments are both learned from the coarse-to-fine regression, which aims to map features to the shape increment and pose residuals. Therefore, JFA jointly learns the head pose estimation and face alignment by leveraging the global and local CNN features. Note that the most similar work to our method is HyperFace, proposed by Ranjan *et al.* [18]. The authors treat the face detection, landmark localization, pose estimation, and gender recognition in one end-to-end framework. However, HyperFace only explores the global features and does not support a fine process for the pose and landmarks. JFA is designed in a hierarchical way, which analyzes the face from global to local in a cascade manner. It uses global CNN features to provide better initialization, which reduces the variation from the realistic bounding box. The local CNNs provide the discriminative features for the cascaded regression.

In summary, the contributions are as follows:

1) We leverage the relationship between head pose and landmarks to search for the best shape for initialization.
2) To the best of our knowledge, this is the first work to explore the deep global and local features together via CNNs on the joint head pose estimation and face alignment in a cascaded way.

The rest of the paper is organized as follows: a brief literature review is presented in Sec. II. In Sec. III, we introduce our system step-by-step. Subsequently, the implementation details such as augmentation and parameters are provided in Sec. IV. Finally, we present a number of evaluations and demonstrate the efficiency, accuracy, robustness, and generality of our proposed novel algorithm. The supplementary meta-data used for training are available at cbl.uh.edu/repository-data.

II. Literature Review

A. Head pose estimation

Head pose estimation is used to infer the orientation of the head relative to the camera. There is limited literature that addresses this topic, and most of it focuses on 3D head pose estimation. We believe that the reason is the lack of data with accurate annotations. Murphy-Chutorian and Trivedi [16] summarize methods before 2009. Geng and Xia [10] proposed multivariate label distribution to soft label the pose that contains neighborhood poses. This method is limited in accuracy by the use of pose grids. Yang *et al.* [28] demonstrated that head pose estimation can benefit the face alignment process by sampling the shape according to the nearest pose neighbor or re-projection from a 3D face model.

B. Face alignment

The modern algorithms in this domain follow the general cascade regression (GCR) framework proposed by Dollar *et al.* [8]. In particular, GCR concatenates many regressors in a cascade chain. For each regressor, it learns a mapping function from the feature domain to the target output (*e.g.*, shape residuals in this problem). Before the regression, different features are learned to represent the current shape. It is shown that the features are discriminative if they are learned from the current shape in the literature [5], [19], [29] called *shape-indexed features*. In addition, some algorithms choose different regressors for the fitting process (*e.g.*, random ferns [5], random forest [13], [15], linear regressor [25], [19], [27], and neural networks [30], [22]). As the last step in the cascade, the shape is updated according to the sum of the current shape and the learned shape increment.

Specifically, Xiong *et al.* [25] developed a method called Supervised Descent Method (SDM) by investigating linear regression with strong hand-crafted features such as SIFT. Asthana *et al.* [2] proposed a discriminative response map fitting (DRMF) process for the face alignment task using a Response Patch Model learned from PCA. Burgos-Artizzu *et al.* [4] divided the face into nine regions and proposed Robust Cascade Pose Regression (RCPR) by adding the classification of the occlusion area. The incremental face alignment method (Chehra) [3] can incrementally update the linear regressors in parallel by addressing the re-training problem of sequence learning when the new samples arrive. Ren *et al.* [19] proposed learning local binary features by using random forests and demonstrated that it achieves as fast as 3,000 FPS. In addition, an ensemble of regression trees was used by Kazemi *et al.* [13] to localize the face landmarks. A Gauss-Newton Deformable Part Model (GN-DPM) was proposed by Tzimiropoulos *et al.* [24], treating the model fitting as the Gauss-Newton process. Zhao *et al.* [31] combined the pose, expression, and shape in a unified multi-output framework using the random forest. This framework concatenated the pose estimation, expression prediction, and landmark detection in a sequence and maximized an energy function that measured each task interactively. Xiong and Torre [26] extended SDM into a global supervised descent method (GSDM) by dividing the Homogenous Descent into several parts. The project-out cascade regression (POCR) proposed by Tzimiropoulos [23] applied the parametric generative model for both shape and appearance. Zhu *et al.* [32] combined the exemplar searching and regression method together, searching for similar shapes from a shape pool using a probabilistic function. Xu *et al.* [27] proposed an initialization method based on part detection and used random ferns to learn features. Zhu *et al.* [33] learned the random forest to choose a homogeneous domain to optimization, each of which was handled by a regressor. However, most of these works only focus on the local patches and start from the mean shape, which would easily be stuck at a local minimum.

Deep learning methods are also employed in this domain. Sun *et al.* [21] learned three-level cascade convolutional networks to fit five landmarks. Zhang *et al.* [30] applied Auto-encoder networks (CFAN) that combined several stacked auto-encoder networks with different resolution images. Tri-

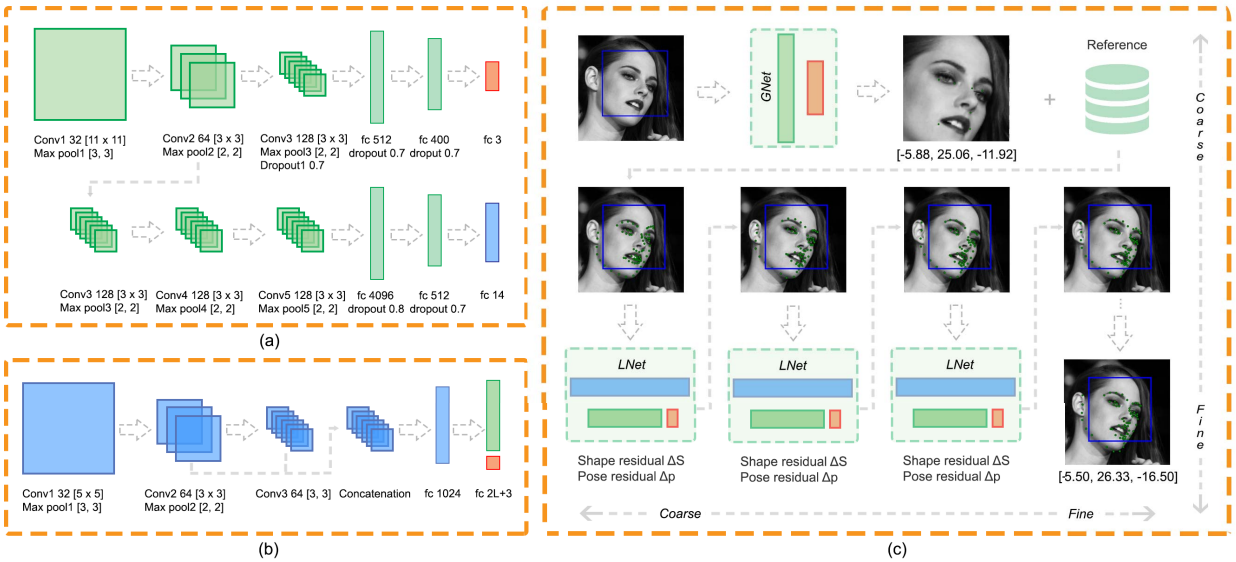


Fig. 2: Overview of proposed JFA system. (a) Depiction of the *GNet* structure for initial head pose estimation and primary landmark estimation. (b) Depiction of the *LNet* structure for feature extraction. (c) Overview of the joint head pose estimation and face alignment learning. *GNet* is used to predict the head pose estimation and initial shape. From the pose distribution, the most similar pose and the corresponding shape are selected as the initialization for joint residuals learning. *LNet* is used to learn the non-linear feature representation from the extracted patches according to the current shape. Then, the regressors are learned from the local CNN features to the shape residual and pose residual, which are used to update the shape for the next iteration. The system is designed based on coarse-to-fine principles exploring global and local features.

georgis *et al.* [22] learned the features using CNN and fit the features to the Recurrent Neural Network (RNN). They demonstrated that the CNN-learned features are more discriminant than hand-crafted features such as SIFT. In addition, a 3D model fitting process using deep learning was also proposed recently which overcomes the large pose and self-occlusion problem. Jourabloo and Liu [11] applied a 3D Morphable Model (3DMM) to this task and updated a projection matrix as well as shape parameters using a sequence of CNN regressors. Another 3D fitting process called 3DDFA was proposed by Zhu *et al.* [34], who enlarged the data by 3D image meshing and rotation and learned shape parameters by CNN. In summary, the differences between the methods above are in these four aspects: (i) initialization, (ii) features, (iii) regressors, and (iv) framework design.

III. Method

Given a face image $\mathbf{I}$, 2D face alignment is the task to localize L pre-defined fiducial points, $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_L]$, within an image, where $\mathbf{s}_i$ denotes the coordinates of i^{th} landmark. The common cascade shape regression approach updates the shape according to the following function in t iteration:

$$\mathbf{S}^t = \mathbf{S}^{t-1} + \mathbf{W}^t \Phi^t(\mathbf{I}, \mathbf{S}^{t-1}), \tag{1}$$

where $\mathbf{W}$ is the weight learned from the training procedure and $\Phi(\cdot)$ is the function to extract features according to the current shape. In training, the algorithm always starts from a mean shape and learns a weight matrix mapping from current shape to shape increment.

In this section, we introduce in detail the joint head pose estimation and face alignment system (JFA), which iteratively learns regressors for the head pose and landmark detection tasks extended from the cascade regression.

A. System design

Head pose and face alignment are highly correlated. We used the head pose distributions from the reference database and learned CNN features to reduce errors in head pose estimation and face alignment tasks jointly.

In Fig. 2, we illustrate the JFA pipeline. Given a face image, the off-the-shelf Dlib [14] face detector is used to locate the bounding box of the face. The first part of our system is *GNet* (Fig. 2(a)), which estimates head pose and facial landmarks using global CNN features. With the predicted head pose, the probabilities of reference shapes are computed according to the pose. The initial shape is generated by selecting the reference shape with the highest probability and aligning to the predicted shape. The next step is a coarse-to-fine regression for the pose and landmarks using *LNet* (Fig. 2(b)). The patches are extracted according to the current shape and are propagated through *LNet* to obtain the non-linear local CNN feature representations for pose and shape. The local CNN features are employed to learn the shape and pose residuals by linear projection. The shape and pose residuals are added to the current shapes

to update for the next iteration. The training procedure of JFA is summarized in Alg. 1. The system is designed in a hierarchical way based on coarse-to-fine principles, which refines the shape and pose sequentially.

B. GNet: Initialization

Previous works such as SDM [25], LBF [19], and ERT [13] always use a mean shape for the initialization. However, it has been shown that landmark distribution is highly correlated to head pose due to pose variation [28]. Therefore, we explore how to obtain a precise initialization compared with a mean shape for the following procedures.

We designed *GNet* to explore the global information from the whole face image to estimate the initial head pose and landmarks given a set of images $\{\mathbf{I}_i\}$ and corresponding detected bounding box $\{\mathbf{b}_i\}$ as well as ground-truth poses and shapes $\{\bar{\mathbf{p}}_i, \bar{\mathbf{S}}_i\}$, where $i = 1, \ldots, N$. Here, we treat the head as a 3D object and its orientation can be represented by three angles: pitch, yaw, and roll. For the global prediction, we only extract the face area using the predicted face bounding box to avoid the background. However, the bounding box contains a part of a face, which does not always contain the cheek area (Fig. 1). Therefore, we designed the system in a hierarchical way to solve this problem. First, we detect the common landmarks within the face region, then search for a similar shape to be used as an initialization.

We define seven primary landmarks including the corners of the eyes, the nose tip, and the corners of the mouth. This process can be treated as facial component localization. As illustrated in Fig. 2(a), *GNet* consists of two CNN sequences which share the first two convolutional layers. The first sequence is used to predict the head pose while the second one is used to localize the initial primary landmarks. The head pose estimation sequence has one additional convolutional layer to extract the middle-level feature to predict the head pose. The second CNN contains another three convolutional layers to explore low-level features for localizing the initial primary landmarks.

The loss function used in training is defined as follows:

$$l_1 = \frac{1}{N_b}\sum_{i=1}^{N_b} ||\mathbf{p} - \bar{\mathbf{p}}||_2^2 + \frac{\lambda}{N_b * L}\sum_{i=1}^{N_b} ||\mathbf{S} - \bar{\mathbf{S}}||_2^2, \qquad (2)$$

where N_b is the number of mini-batch feeds in training, and λ denotes the weight to balance the contributions of the two terms.

We use *GNet* to predict the head pose and initial landmarks for both the training set and testing set. Given an estimated pose $\mathbf{p}^*$ and initial shape $\mathbf{S}^*$, we compute the probability of shapes in a reference shape pool and choose the one with the highest probability as the initialization for that face. During training, it is possible that the shape selected does not need to be modified. In that situation, another shape is selected so that the corresponding pose is close to the estimated pose. However, it is not enough that facial shapes $\mathbf{S}$ in the pose domain are similar. The shape assigned on the 2D image plane may have large variations in translation and scale. We take advantage of the facial component locations by solving a linear square equation between $\mathbf{S}^*$ and $\mathbf{S}$ (note that here $\mathbf{S}$ denotes the corresponding points in $\mathbf{S}$) to obtain the affine transformation matrix $\mathbf{M}$. Then, the transformed shape $\mathbf{S}^0 = \mathbf{S} * \mathbf{M}$ is used as shape initialization. The initial shape $\mathbf{p}^0$ is directly set to $\mathbf{p}^*$.

Algorithm 1 The training procedure of JFA.

Input: Images, bounding boxes, ground-truth poses and shapes $\{\mathbf{I}_i,\ \mathbf{b}_i,\ \bar{\mathbf{p}}_i,\ \bar{\mathbf{S}}_i\},\ i \in \{1, \ldots, N\},\ \%\ T$ is the number of cascades
Output: JFA models
Procedure:

```
Train GNet using the entire face region
Predict head pose and initial shape using GNet
Search the shape S^0 according to the pose
for t = 1 to T do
    Extract patches on images
    Train LNet to learn local CNN features
    Regress ΔS^t and Δp^t
    Update the shape S^{t+1} ← S^t + ΔS^t
    Update the pose p^{t+1} ← p^t + Δp^t
    if t < T then
        Perturb S^t by searching for a similar pose
    end if
end for
```

C. LNet: Feature extraction and Regression

Based on the observation that CNN features are more discriminative than conventional features, we designed *LNet* to obtain the local CNN features from local patches. *LNet* has three convolutional layers, two max-pooling layers, and two fully-connected layers. Without losing information, the feature maps generated by the second and third convolutional layers are concatenated and are connected to a fully-connected layer to control the length of features. The loss function in *LNet* is defined to minimize the difference between the predicted and ground-truth residuals as follows:

$$l_2 = \frac{1}{N_b}\sum_{i=1}^{N_b} ||\Delta\mathbf{p}^t - \Delta\bar{\mathbf{p}}^t||_2^2 + \frac{\lambda}{N_b * L}\sum_{i=1}^{N_b} ||\Delta\mathbf{S}^t - \Delta\bar{\mathbf{S}}^t||_2^2, \qquad (3)$$

where $\Delta\mathbf{p}^t$ and $\Delta\mathbf{S}^t$ denote the predicted pose and shape residuals in t iteration, respectively.

Two linear regressions are used to predict the shape increment and pose increment. With the predictions $\Delta\mathbf{p}^t$ and $\Delta\mathbf{S}^t$ in t iteration, the pose and shape are updated by $\mathbf{p}^{t+1} = \mathbf{p}^t + \Delta\mathbf{p}^t$ and $\mathbf{S}^{t+1} = \mathbf{S}^t + \Delta\mathbf{S}^t$.

IV. Implementation Details

A. Data preparation

We use the datasets from the 300-W competition [22] including LFPW, AFW, HELEN, and IBUG, which are widely used in recent face alignment methods. The common configuration for 300-W has three parts when the testing set of 300-W competition is not publically available.

Training set: The training set consists of the training set from LFPW and HELEN, as well as the whole AFW, the

Method	Face detector	51 landmarks					68 landmarks				
		LFPW	HELEN	*Common*	*Challenge*	*Full*	LFPW	HELEN	*Common*	*Challenge*	*Full*
DRMF [2]	MATLAB	4.95	6.11	5.64	14.82	7.44	5.80	7.26	6.67	16.66	8.63
Chehra [3]	MATLAB	4.10	4.95	4.60	15.83	6.80	-	-	-	-	-
LBF [19]	OpenCV	4.63	5.69	5.26	18.58	7.87	5.58	6.58	6.18	18.94	8.68
ERT [13]	Dlib	**3.81**	**4.04**	**3.94**	12.17	**5.55**	**4.59**	**4.96**	**4.81**	13.66	6.55
3DDFA* [34]	Dlib	66.64	13.03	34.71	28.60	33.51	-	-	-	-	-
JFA	Dlib	4.65	5.26	5.01	**8.98**	5.79	5.08	5.48	5.32	**9.11**	**6.06**

TABLE I: Comparison of *NMRSE* from different state-of-the-art approaches and corresponding face detector on 300-W *Common set*, *Challenge set*, and *Full set* (% is dropped for simplicity). The detailed experimental settings are described in Sec. V-A. ERT is retrained with Dlib bounding box and the others are used with the pre-trained model. For those images that the face detector fails, we apply bounding box regression to provide a similar bounding box. 3DDFA* is computed using 27 landmarks from MPEG4 index.

total of which is 3,148 images. Specifically, 811 images are from LFPW, 2,000 images are from HELEN, and the last 337 images are from AFW.

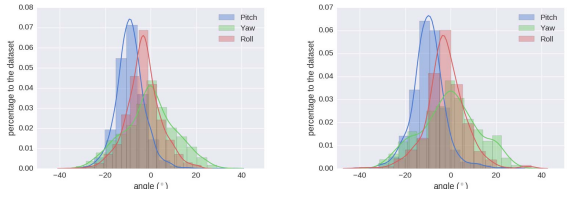
Common testing set: This testing set is collected from the testing set of LFPW and HELEN. The total of the testing set is 689 images, 224 images of which are from the LFPW testing set.

Challenge testing set: The challenge testing set is IBUG data, which has 135 images. It is very challenging because the images have larger variations of pose and lower resolutions compared to the training set.

The combination of *common testing set* and *challenge testing set* is called the *full testing set*. These datasets are annotated with 68 landmarks but without the pose information. To enlarge the meta-data, we run the pose estimator by Asthana *et al.* [2] to estimate the head pose. In addition, to obtain a face bounding box similar to the requirement of the state-of-the-art face alignment system, we apply Dlib, OpenCV, and MATLAB face detectors for the preparation of the baselines. For those images for which a face cannot be detected, we apply a bounding box regression from the 300-W tiny bounding box to provide the estimated bounding box. Recently, the *test set* from 300-W competition became publicly available. Due to lack of data, we use it as the validation set.

B. Augmentation

To successfully train the JFA and to avoid over-fitting, we apply the data augmentation to obtain a large number of training samples. We randomly re-size the bounding box with zoom factor $\in [0.9, 1.1]$ and translate the bounding box. In addition to six times scaling and perturbing the bounding box, we also rotate the images slightly to make the distribution of the roll variable larger. From our experiment of distribution of the head pose (Fig. 3), we observe that the head pose distribution of the *full testing set* is wider than the *training set*. To decrease the distribution variation between two sets, we conduct the rotation operation according to the head pose. From our experience, the probability of rotation is small when it has a large pitch angle or roll angle. Therefore, we use a multivariate Gaussian distribution to model this effect and only rotate the images which have a high probability of fitting the model. This operation widens the angle distribution and decreases the variation conducted by a rotation operation.



(a) (b)

Fig. 3: The head pose distribution of datasets: (a) *training set*, (b) *full testing set*. The *full testing set* has a wider pose distribution than the *training set*, especially on roll.

C. Bounding box regression

The current off-the-shelf face detectors (*e.g.*, Dlib) still cannot achieve a 100% detection rate on 300-W datasets. For the pictures that the face detector fails, a random forest regression from the anchor bounding box is being learned, which is set as the 300-W compact bounding box $[x_1^a, y_1^a, x_2^a, y_2^a]$, to the detected face bounding box. We use the overlap k of the detected bounding box with the compact bounding box (in our experiments, $k = 0.35$) to avoid an incorrect bounding box. For the regression, we set the target of four dimensions as follows:

$$\alpha = \frac{x_1^* - x_1^a}{x_2^a - x_1^a}, \beta = \frac{y_1^* - y_1^a}{y_2^a - y_1^a}, \gamma = \frac{x_2^* - x_2^a}{x_2^a - x_1^a}, \theta = \frac{y_2^* - y_2^a}{y_2^a - y_1^a}, \quad (4)$$

where $[x_1^*, y_1^*, x_2^*, y_2^*]$ denotes the coordinates of the bounding box generated from the face detector. A random forest regression using HoG features is learned to map HoG features from a 300-W compact bounding box to the bounding box that is similar to the one that a face detector generates.

D. Other parameters

In training, we normalize the data. *GNet* is trained on the gray-scale 96×96 images augmented by slightly translating the bounding box five times and the target poses are normalized to $[-1, 1]$. *LNet* uses gray-scale images, which are re-sized to 224×224 and the patch size is set to 20×20. The balance variable λ in the total loss function is set to 1

Fig. 4: Selected landmark detection results on *challenge testing set*. (T) Depicted are the results from ERT (Dlib implementation); (B) Depiction of the JFA results.

and 10 in *GNet* and *LNet*, respectively. Batch normalization is used before the first convolutional layer in the *LNet* to accelerate the training process. The activation *relu* is applied only on the first convolutional layer and linear activation is used for other convolutional layers. The initial learning rate is 0.001 and decreases with exponential decay of 0.1 every 8,000 steps.

V. Experimental Design

A. Landmark detection

We compare our method against state-of-the-art methods in two types of experiments. To provide a fair and intuitive comparison, we first evaluate our face alignment on the *full testing set* from 300-W including the *common testing set* and *challenge testing set*. The baselines consist of various state-of-the-art methods including ERT [13], LBF [19], Chehra [3], DRMF [2], and 3DDFA [34]. For a fair comparison, we re-train ERT using the same bounding box that we use. The ERT algorithm is implemented by Dlib and we re-train the model with its default setting. We used an implementation[1] and their pre-trained model of LBF with OpenCV face detector. Additionally, we prepare the bounding box provided by MATLAB face detector for Chehra and DRMF. 3DDFA [34] is tested using the Dlib bounding box. The MPEG4 index is used to extract the landmarks provided by Paysan *et al.* [17] (27 points are selected to correspond to the ground-truth annotations). We believe this setting is reasonable because we provide a bounding box similar to the one each algorithm used in its training process. We also implemented a deep learning method such as Hyperface and Multitask face described by Ranjan [18] using the same data. However, the performance we got is not comparable to the conventional methods so we do not report the results here. Some possible reasons include: (1) since the notion of bounding box is not used, the image contains a large region of background; (2) the number of subjects in the database is small, thus CNN over-fits in the end-to-end training. In our system, *GNet* is learned from the face region and locates the facial primary landmarks so that it reduces the variance from the background. In the cascaded local patch learning, the shapes are re-generated to avoid over-fitting to some extent.

To compare with face alignment results, we adopt the metric from the 300-W competition, which computes the normalized mean root square error (*NMRSE*) defined as follows:

$$NMRSE = \frac{1}{L * d} \sum_{i=1}^{L} ||\mathbf{s} - \bar{\mathbf{s}}||_2^2, \tag{5}$$

where d is the distance between the two outer eye corners. Moreover, we provide the evaluation on inner face alignment, which excludes the outer contour of the face (17 landmarks). Different results are obtained using 51 inner face landmarks and 68 full landmarks. We believe this is because the contour has much more variation and is not consistent. For a full evaluation of the methods, we report the *NMRSE* results for both conditions, which are summarized in Table I. We drop the notation % for simplicity.

Although JFA is not prominent on the common pose face images contained in LFPW and Helen, it outperforms the conventional methods such as ERT by 33% on the face images with a large pose. This is because the pose information is explored in *GNet* and the similar shape is selected as initialization for the cascade local pose and shape refinement using *LNet*.

B. Head pose estimation

First, we compare the head pose with the work by Yang *et al.* [28] measured by absolute mean error (*AME*, defined

[1] https://github.com/yulequan/face-alignment-in-3000fps

in Eq. 6) of three dimensions: pitch, yaw, and roll:

$$AME = \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{p} - \bar{\mathbf{p}}||_1. \quad (6)$$

Both methods were trained on the training set of 300-W and tested on the *full set*. The difference is that we adopt a different pose estimator. To obtain a fair comparison, we adopt their result directly from the literature. Then, we compare the learned features with the traditional method. Using the same augmented training data with our method, we trained the model using random forest and super vector regression, respectively, using HoG features. The random forest is set to contain 100 trees. The cell size of the HoG extractor is set to 8×8 with nine cells in the same block.

The results are listed in Table II. Compared with a conventional random forest approach such as random forest, our method can boost the accuracy of 54% in total: 36% on the pitch, 55% on the yaw, and 46% on the roll. The result implies that a deep learning method outperforms the conventional methods in this task.

TABLE II: Comparison of *AME* for head pose estimation

Method	Pitch	Yaw	Roll
Yang *et al.* [28]	5.1	4.2	**2.4**
Random forest	4.7	5.5	4.8
SVR	4.8	7.8	5.3
GNet	3.5	3.3	2.6
JFA	**3.0**	**2.5**	2.6

C. Self Evaluation

1) Global vs Local: We modified *GNet* and re-trained the network to map the global CNN features to head pose estimation and 68 landmark detection tasks directly without local cascade CNN learning. The final result tested on *full testing set* is around 12%. The result points to the necessity of exploring the local features.

2) Number of cascades: In Fig. 5, the shape loss *NMRSE* on the 300-W *full testing set* is reported with increasing the length of cascade of JFA. The *NMRSE* tends to converge after four iterations.

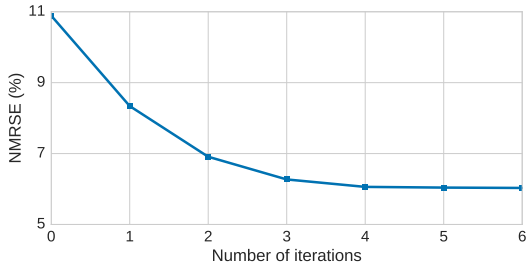


Fig. 5: The testing error with increasing the number of cascades.

3) Time Complexity Evaluation: Our framework is implemented in Python and runs on Desktop with Intel i7-6700K CPU and GTX 960 Graphic card. The average time for running *GNet* is about 0.8 *ms* per sample. *LNet* consumes 5 *ms* for one iteration. The total average processing time for JFA is around 21 *ms* per sample including the time for loading the model.

VI. Discussion and Future Work

The experimental analysis offers insights for future research:

1) A hierarchical face analysis system can analyze face attributes on different levels. Global CNN features can handle global information such as head pose easily, but are not suitable for detailed information such as landmarks. The local CNN features are used to refine shapes and poses.
2) There is a critical need to obtain a new facial databases containing more data and subjects to be used with the deep learning methods. Although *AME* on the head pose estimation is less than 3°, JFA is trained on the estimated data. For precise head pose estimation, it requires accurate data.
3) In our experiments, the cascaded *LNet* fails to detect the landmarks in the common pose images compared with the ERT algorithm. One possible reason is that the small number of subjects influence the performance. We will explore this issue in the future.

VII. Conclusion

In this paper, we addressed head pose estimation and landmark detection. We proposed a joint hierarchical head pose estimation and face alignment learning system exploring the global and local CNN features. First, *GNet* is trained on the detected face region to obtain a rough estimate of pose and localize the seven primary landmarks. The most similar shape is selected from a reference as initialization. Then, *LNet* is used to learn local CNN features and predict the shape and pose residuals. Based on the coarse-to-fine manner, the global CNN features are used to estimate face attributes such as head pose and facial components while local CNN features are used to refine the shape in the cascade. To the best of our knowledge, this is the first system that explores the global and local CNN features to solve head pose estimation and landmark detection tasks jointly. Our experiments demonstrate that our system outperforms conventional head pose estimation on the challenging head pose estimation task.

VIII. Acknowledgment

This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number 2015-ST-061-BSH001. This grant is awarded to the Borders, Trade, and Immigration (BTI) Institute: A DHS Center of Excellence led by the University of Houston, and includes support for the project “Image and Video Person Identification in an Operational Environment” awarded to the

University of Houston. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

REFERENCES

[1] B. Amos, B. Ludwiczuk, and S. Mahadev. Openface: A general-purpose face recognition library with mobile applications. Technical Report CMU-CS-16-118, CMU School of Computer Science, Pittsburgh, PA, 2016.

[2] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Robust discriminative response map fitting with constrained local models. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 3444–3451, Portland, Oregon, June 23-28 2013.

[3] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Incremental face alignment in the wild. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1859–1866, Columbus, OH, June 2014.

[4] X. P. Burgos-Artizzu, P. Perona, and P. Dollár. Robust face landmark estimation under occlusion. In *Proc. IEEE International Conference on Computer Vision*, pages 1–8, Sydney, Australia, December 3-6 2013.

[5] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2887–2894, Providence, RI, June 16-21 2012.

[6] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, January 1995.

[7] D. Cristinacce and T. Cootes. Feature detection and tracking with constrained local models. In *Proc. British Machine Vision Conference*, pages 929–938, Edinburgh, UK, Sep.4-7 2006.

[8] P. Dollar, P. Welinder, and P. Perona. Cascaded pose regression. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1078–1085, San Francisco, CA, Jun. 13-18 2010.

[9] P. Dou, Y. Wu, S. Shah, and I. Kakadiaris. Robust 3D facial shape reconstruction from single images via two-fold coupled structure learning. In *Proc. British Machine Vision Conference*, pages 1–13, Nottingham, United Kingdom, September 1-5 2014.

[10] X. Geng and Y. Xia. Head pose estimation based on multivariate label distribution. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1837–1842, Columbus, OH, June 24-27 2014.

[11] A. Jourabloo and X. Liu. Large-pose face alignment via CNN-based dense 3D model fitting. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, June 26-July 1 2016.

[12] I. A. Kakadiaris, G. Passalis, G. Toderici, M. N. Murtuza, Y. Lu, N. Karampatziakis, and T. Theoharis. Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):640–649, 2007.

[13] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867 – 1874, Columbus, OH, June 24-27 2014.

[14] D. E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[15] D. Lee, H. Park, and C. D. Yoo. Face alignment using cascade gaussian process regression trees. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 4204–4212, Boston, Massachusetts, June 7 - 12 2015.

[16] E. Murphy-Chutorian and M. Trivedi. Head pose estimation in computer vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):607–626, Apr. 2009.

[17] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3D face model for pose and illumination invariant face recognition. In *Proc. 6th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 296–301, Genoa, Italy, Sep. 2-4 2009.

[18] R. Ranjan, V. M. Patel, and R. Chellappa. HyperFace: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition, May 2016.

[19] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3,000 FPS via regressing local binary features. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, June 24-27 2014.

[20] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: database and results. *Image and Vision Computing*, 47:3 – 18, March 2016.

[21] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *Proc. Computer Vision and Pattern Recognition*, Portland, Oregon, June 25-27 2013.

[22] G. Trigeorgis, P. Snape, M. A. Nicolaou, E. Antonakos, and S. Zafeiriou. Mnemonic descent method: A recurrent process applied for end-to-end face alignment. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 4177–4187, Las Vegas, NV, June 26 - July 1 2016.

[23] G. Tzimiropoulos. Project-out cascaded regression with an application to face alignment. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 3659–3667, Boston, Massachusetts, June 7-12 2015.

[24] G. Tzimiropoulos and M. Pantic. Gauss-Newton deformable part models for face alignment in-the-wild. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, Columbus, OH, June 24-27 2014.

[25] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 532–539, Portland, Oregon, June 25-27 2013.

[26] X. Xiong and F. D. la Torre. Global supervised descent method. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2664–2673, Boston, Massachusetts, June 7 - 12 2015.

[27] X. Xu, S. Shah, and I. A. Kakadiaris. Face alignment via an ensemble of random ferns. In *Proc. IEEE International Conference on Identity, Security and Behavior Analysis*, Sendai, Japan, Feb. 29 - Mar. 2 2016.

[28] H. Yang, W. Mou, Y. Zhang, I. Patras, H. Gunes, and P. Robinson. Face alignment assisted by head pose estimation. In *Proc. 26th British Machine Vision Conference*, Swansea, UK, September 7-10 2015.

[29] H. Yang, R. Zhang, and P. Robinson. Human and sheep facial landmarks localisation by triplet interpolated features. In *Proc. IEEE Winter Conference on Applications of Computer Vision*, Lake Placid, NY, March 7-9 2016.

[30] J. Zhang, S. Shan, M. Kan, and X. Chen. Coarse-to-fine auto-encoder networks (CFAN) for real-time face alignment. In *Proc. 13th European Conference on Computer Vision*, pages 1–16, Zurich, Switzerland, Sep. 6-12 2014.

[31] X. Zhao, T.-K. K., and W. Luo. Unified face analysis by iterative multi-output random forests. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1765–1772, Columbus, OH, June 23-28 2014.

[32] S. Zhu, C. Li, C. C. Loy, and X. Tang. Face alignment by coarse-to-fine shape searching. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 4998–5006, Boston, MA, June 7 - 12 2015.

[33] S. Zhu, C. Li, C. C. Loy, and X. Tang. Unconstrained face alignment via cascaded compositional learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, June 26-July 1 2016.

[34] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li. Face alignment across large poses: A 3D solution. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, June 26-July 1 2016.

[35] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2879–2886, Providence, RI, June 16-21 2012.