# FSA-Net: Learning Fine-Grained Structure Aggregation for Head Pose Estimation from a Single Image

Tsun-Yi Yang[1,2] Yi-Ting Chen[1] Yen-Yu Lin[1] Yung-Yu Chuang[1,2]
[1]Academia Sinica, Taiwan [2]National Taiwan University, Taiwan
shamangary@citi.sinica.edu.tw jamie@media.ee.ntu.edu.tw yylin@citi.sinica.edu.tw cyy@csie.ntu.edu.tw

## Abstract

*This paper proposes a method for head pose estimation from a single image. Previous methods often predict head poses through landmark or depth estimation and would require more computation than necessary. Our method is based on regression and feature aggregation. For having a compact model, we employ the soft stagewise regression scheme. Existing feature aggregation methods treat inputs as a bag of features and thus ignore their spatial relationship in a feature map. We propose to learn a fine-grained structure mapping for spatially grouping features before aggregation. The fine-grained structure provides part-based information and pooled values. By utilizing learnable and non-learnable importance over the spatial location, different model variants can be generated and form a complementary ensemble. Experiments show that our method outperforms the state-of-the-art methods including both the landmark-free ones and the ones based on landmark or depth estimation. With only a single RGB frame as input, our method even outperforms methods utilizing multi-modality information (RGB-D, RGB-Time) on estimating the yaw angle. Furthermore, the memory overhead of our model is* $100\times$ *smaller than those of previous methods.*

## 1. Introduction

Facial modeling and analysis have long been an active research topic in computer vision [2, 3, 4, 5, 6, 7, 21, 22, 24, 25]. Large facial datasets [16, 37, 48] and efficient methods for different facial analysis problems have been proposed for years, such as face recognition [4, 6] or identification, facial age estimation [45], landmark detection [3], and head pose estimation [35]. This paper addresses the head pose estimation problem which has many applications such as driver behavior monitoring and human attention modelling. It could also be used to improve or provide extra information for other problems such as identity recognition [39], expression recognition [46], or attention detection [8].

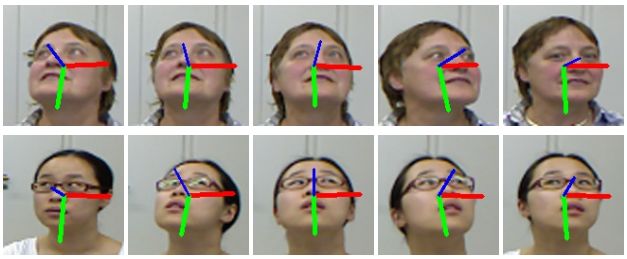


Figure 1. Sample results of pose estimation using the proposed method. Our method only takes as input a single RGB frame. Results for two sequences of head motion are shown. The blue line indicates the direction the subject is facing; the green line for the downward direction while the red one for the side.

Head pose estimation from a single image is a challenging problem. Head pose is a 3D vector containing the angles of yaw, pitch and roll. Estimating the head pose from an image essentially requires to learn a mapping between 2D and 3D spaces. Some methods utilize more modalities such as 3D information in depth images [28, 25, 14, 27] or temporal information in video sequences [16]. The depth images provide 3D information that is missing in 2D images. Videos capture continuous movement of human heads and provide extra information to help the pose estimation. However, learning the temporal information is usually achieved by recurrent structures with high computation costs while capturing depth information often requires special cameras which are not always available. Most single-frame pose estimation methods utilize facial landmark detection for estimating head poses [20, 3]. However, it would incur more computation and leads to bigger models. Hence, all these models are not suitable to be adopted on platforms with limited memory and computation resource.

This paper proposes FSA-Net, a compact model for pose estimation from a single image using direct regression without landmarks. For having a compact model, the proposed model is built on the soft stagewise regression scheme [45]. To harvest multi-scale information, like many regression methods [45, 3], our method combines feature maps from different layers/stages. For having more accurate predic-

tions, it requires to learn meaningful intermediate features for performing regression. The state-of-the-art differentiable aggregation/pooling methods such as capsule networks [36] and NetVLAD [1] can be adopted for distilling representative features from candidate features. However, these methods often treat the inputs as a bag of features and neglect their spatial relationship in the feature map. The key idea of the proposed method is to spatially group pixel-level features of the feature map together into a set of features encoded with spatial information. These features are then used as the candidate features for aggregation. That is, the proposed method learns to find the fine-grained structure mapping for spatially grouping pixel-level features together to form more power region-level features.

The proposed fine-grained structure mapping can be interpreted as a more flexible and versatile tool for pooling. Conventional pooling takes a set of features at fixed locations within a local window. A pre-defined operation is applied to them without taking data content into account while our method pools features from a wider area with a more versatile operation. For harvesting more versatile spatial information, we adopt both learnable and non-learnable importance measures, and complementary model variants can be generated for making a powerful and robust ensemble. Experiments show that our model outperforms other single-frame pose estimation methods with the model size of only 5MB, around $100\times$ smaller than that of the previous state-of-the-art method. For yaw angle prediction, the proposed method is even favorable against heavy models utilizing multiple modalities such as RGB-D or RGB-Time. Figure 1 shows sample results of the proposed method. It is clear that the pose estimation is rather accurate.

## 2. Related work

**Landmark-based methods.** They find facial landmarks first and then use them to estimate the head pose. Given a set of 2D face landmarks, the head pose can be determined by 3D computer vision techniques such as POSIT [11]. Regression-based methods [5, 43, 12, 23, 42] sketch initial faces, and incrementally align the drawn faces to real ones by regression. Model-based methods [26, 24, 10] model human faces with several key points, and then locate the key points on real faces via trained appearance models. Deep-learning-based methods [48, 3, 38] estimate 3D face models using convolutional neural networks (CNNs) and gain superior performance compared to previous methods. Although effective, landmark detection is not required for pose estimation and often incurs unnecessary computation.

**Methods with different modalities.** Landmark-based methods require manually annotated labels as ground truth. However, acquiring annotated landmarks is labor-intensive. In some cases with low-resolution images, even experts cannot accurately pinpoint facial landmark locations. Considering the cost and accuracy, some proposed face alignment algorithms without face landmarks [6, 35]. On the other hand, it is also very popular to adopt different modalities to compensate for the loss of information [14, 27, 9, 29].

*RGB.* Several approaches only utilize a single RGB image for pose estimation [6, 35, 32, 22, 31]. FacePoseNet [6] employed a CNN for 3D head pose regression, which improves face recognition accuracy. Nataniel *et al*. [35] combined ResNet50 with a multi-loss architecture. Each loss contains a binned pose classification and regression, corresponding to yaw, pitch, and roll individually. With binned classification, their method obtained robust neighborhood prediction of the pose.

*Depth.* Intensity-based head pose estimation algorithms fail to produce accurate head poses in conditions such as poor illumination during night time or large illumination variations during day time. Fanelli *et al*. [14] exploited discriminative random regression forests for head pose estimation with depth images. Meyer *et al*. [27] proposed to register 3D morphable models to depth images and incrementally refine the registration over time.

*RGB+Time.* Methods for facial video analysis take a sequence of RGB images as inputs and utilize temporal information. Previous facial analysis methods on videos [9, 29] cope with temporal coherence by Bayesian filters or particle filters. Inspired by the similarity between Bayesian filters and recurrent neural networks (RNNs), Gu *et al*. [16] proposed to track facial features by RNNs over time.

**Multi-task methods.** Head pose estimation is closely related to other facial analysis problems. Recent work [7, 31, 49] demonstrates that learning related tasks jointly achieves better results than performing individual tasks independently. Several methods [31, 32] propose to perform various related facial analysis tasks simultaneously using CNNs. Hyperface [31] learns common features by CNNs, for simultaneously performing face detection, facial landmark localization, head pose estimation and gender recognition. KEPLER [22] learns global and local features by a Heatmap-CNN to explore structural dependencies.

**Attention.** Our method provides attention for pose estimation. Our attention can be optimized in an end-to-end manner along with the pose estimation without complex additional techniques [18, 19, 40, 30, 17]. Compared with other pooling methods using attention such as CBAM [41] and Attentional Pooling [15], our method has the following differences with them. First, they focus on categorical classification problems (image classification and action recognition) while our method is for a regression problem. Second, they only generate one or two spatial heatmaps while our model is capable of generating multiple spatial attention proposals which are more flexible for refining regression values. Finally, our method takes into account multi-scale information, and it's useful to other applications.

## 3. Method

In this section, we first formulate the pose estimation problem (Section 3.1). Next, we introduce the soft stagewise regression and apply it to pose estimation (Section 3.2). We then give an overview of the proposed FSA-Net (Section 3.3). Two important ingredients of the FSA-Net, the scoring function and the fine-grained structure mapping, are then described in Section 3.4 and Section 3.5 respectively. Finally, we explain details of the architecture (Section 3.6).

### 3.1. Problem formulation

For the problem of image-based head pose estimation, we are given a set of training face images $X = \{x_n \mid n = 1, ..., N\}$ and the pose vector $y_n$ for each image $x_n$, where $N$ is the number of images. Each pose vector $y_n$ is a 3D vector whose components respectively correspond to the angles of yaw, pitch, and roll. The goal is to find a function $F$ so that it predicts $\tilde{y} = F(x)$ that matches the real head pose $y$ for the given image $x$ as much as possible. We find $F$ by minimizing the mean absolute error (MAE) between the predicted and the ground truth poses,

$$J(X) = \frac{1}{N}\sum_{n=1}^{N} \|\tilde{y}_n - y_n\|_1 , \tag{1}$$

where $\tilde{y}_n{=}F(x_n)$ is the predicted pose for the training image $x_n$. It is a regression problem by nature.

### 3.2. SSR-Net-MD

Our proposed solution is built on the SSR-Net [45], which provides a compact model for age estimation from a single image. Inspired by DEX [33], SSR-Net casts the regression problem of age estimation as a classification problem by dividing into the age domain into several age classes (bins). A network performs the classification task and outputs a probability distribution for age classes. Given the probability distribution, the age is estimated as the expected value. For having a compact model, SSR-Net adopts a coarse-to-fine strategy for classification. Each stage only performs intermediate classification with a small number of classes, say "relatively younger", "about right" and "relatively older" within the current age group. The next stage refines the decision within the age group assigned by the previous stage [45]. In sum, SSR-Net performs a hierarchical classification and uses the following soft stage-wise regression for estimating the age $\tilde{y}$:

$$\tilde{y} = \sum_{k=1}^{K} \vec{p}^{(k)} \cdot \vec{\mu}^{(k)}, \tag{2}$$

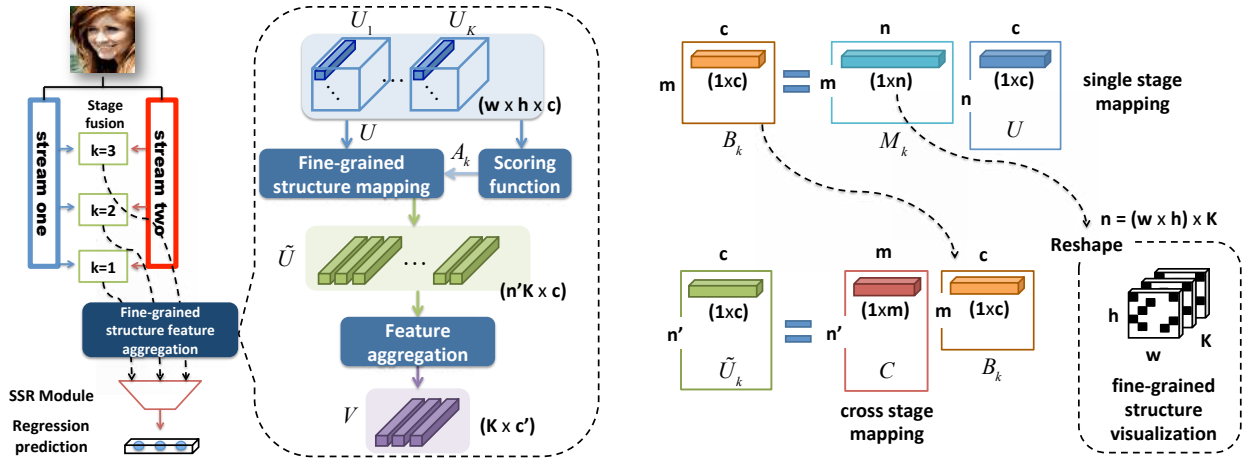
where $K$ is the number of stages; $\vec{p}^{(k)}$ is the probability distribution for the $k$-th stage; and $\vec{\mu}^{(k)}$ is a vector consisting of the representative values of age groups at the $k$-th stage. To accommodate quantization errors and class ambiguity, a shift vector $\vec{\eta}^{(k)}$ adjusts the center for each bin and a scale factor $\Delta_k$ scales the widths of all bins at the $k$-th stage, thus modifying the representative ages $\vec{\mu}^{(k)}$. Like $\vec{p}^{(k)}$, both $\vec{\eta}^{(k)}$ and $\Delta_k$ are found by a neural network. Given an input image, SSR-Net outputs $K$ sets of stage parameters $\{\vec{p}^{(k)}, \vec{\eta}^{(k)}, \Delta_k\}_{k=1}^{K}$ and uses the soft stage-wise regression for estimating the age.

The soft stagewise regression formulation can be applied to any regression problem. For a given regression problem, the soft stagewise regression function $SSR(\{\vec{p}^{(k)}, \vec{\eta}^{(k)}, \Delta_k\}_{k=1}^{K})$ accepts $K$ sets of stage parameters and outputs the expected value as the regression value according to Equation (2). In this paper, we apply the soft stagewise regression to the problem of pose estimation from a single image. Different from the age estimation problem, the pose estimation problem estimates a vector, rather than a scalar. We denote the SSR-Net for multiple dimensional regression by SSR-Net-MD, and revise the two-stream structure of the SSR-Net as described in Section 3.6. Although SSR-Net-MD gives fairly good performance, we propose to use feature aggregation to further improve it.

### 3.3. Overview of FSA-Net

Figure 2(a) depicts the architecture of the proposed FSA-Net. The input image goes through two streams. There are $K$ stages ($K = 3$ in Figure 2(a)). Each stream extracts a feature map at a stage. For the $k$-th stage, the extracted feature maps are fused together by the stage fusion module (the green boxes between two streams in Figure 2(a)). The stage fusion module first combines the two feature maps by element-wise multiplication. It then applies $c$ $1\times1$ convolutions to transform the combined feature map into $c$ channels. Finally, average pooling is used to reduce the size of the feature map to $w\times h$. Thus, we obtain a $w\times h\times c$ feature map $U_k$ for the $k$-th stage. The feature map $U_k$ is a spatial grid, in which each cell contains a $c$-dimensional feature representation of a particular spatial location. These $K$ feature maps are then fed into the mapping module for obtaining $K$ $c'$-d vectors, each of which will be used to obtain the stage outputs $\{\vec{p}^{(k)}, \vec{\eta}^{(k)}, \Delta_k\}$ for the SSR function.

Given $K$ feature maps of size $w \times h \times c$, the task of the aggregation module is to aggregate them into a small number of more representative features, in our case, $K$ $c'$-d features, one for each stage. Through the aggregation process, a more meaningful representation can be distilled from a bag of features. Existing feature aggregation methods, such as capsule [36] and NetVLAD [1], can be employed for the task. However, as mentioned in Section 1, these methods treat the input features as a bag of features and completely ignore the spatial information exhibited within the feature map. To overcome the problem, we propose to perform spatial grouping of features before feeding them into the




(a) FSA-Net (b) Fine-grained structure mapping

Figure 2. Overview of the proposed FSA-Net. Source code available at https://github.com/shamangary/FSA-Net

aggregation process. Thus, the inputs to the feature aggregation module would be more powerful features encoded with global spatial information, rather than the pixel-level features[1] in the feature maps.

For the purpose of spatial grouping, for each feature map $U_k$, we first compute its attention map $A_k$ through a scoring function (Section 3.4). Next, the feature maps $U_k$ and the attention maps $A_k$ are fed into the fine-grained structure mapping module. The module learns to extract $n'$ $c$-d representative features by spatially weighting the pixel-level features in the feature maps. These vectors are then fed into a feature aggregation method for generating the final set of representative features for regression, $V$, containing $K$ $c'$-d features. The vector $V_k$ is used to generate the stage outputs $\{\vec{p}^{(k)}, \vec{\eta}^{(k)}, \Delta_k\}$ for the $k$-th stage through a fully-connected layer. These outputs are then substituted into the SSR function for obtaining the pose estimation.

### 3.4. Scoring function

For better grouping features, it is useful to measure the significance of the pixel-level features. Given a pixel-level feature $u = (u_1, \ldots, u_c)$, we design a scoring function $\Phi(u)$ to measure its importance to facilitate spatial grouping. Thus, for each feature map $U_k$, we obtain an importance or attention map $A_k$, where $A_k(i,j) = \Phi(U_k(i,j))$.

We explore three options as the scoring functions. (1) $1 \times 1$ convolution, (2) Variance and (3) Uniform. The first option takes an extra $1 \times 1$ convolution layer as a learnable scoring function, *i.e.*, $\Phi(u) = \sigma(w \cdot u)$, where $\sigma$ is the sigmoid function and $w$ is the learnable convolution kernel. Although the use of $1\times1$ convolution as the scoring function allows us to learn how to weight features from the training data, it could suffer from the potential overfitting problem when there is significant discrepancy between the training and testing data. Inspired by ORB [34] in which features are selected using variances, the second option explores the use of variance for importance, *i.e.*, $\Phi(u) = \sum_{i=1}^{c}(u_i - \mu)^2$ where $\mu = \frac{1}{c}\sum_{i=1}^{c} u_i$. Note that variance is differentiable although not learnable. The final option, uniform, is to treat all features equally, *i.e.*, $\Phi(u) = 1$. In this case, $\tilde{U} = U$ and the fine-grained structure mapping is not performed. Note that these three options explore learnable, non-learnable and constant alternatives. They could provide complementary information. In Section 4, we will compare the performance of these options. We found that they capture different aspects and the best practice is to form an ensemble model by averaging their predictions together. This way, the pose estimation is more robust.

### 3.5. Fine-grained structure mapping

With the feature maps $U_k$ and their attention maps $A_k$, the next step is to perform fine-grained structure mapping to extract a set of representative features $\tilde{U}$. Figure 2(b) illustrates the process. We first flatten all feature maps $U_k$ into a matrix $U$ whose first dimension is $n = w \times h \times K$, $U \in \mathbb{R}^{n \times c}$. In other words, $U$ is a 2D matrix containing all $c$-d pixel-level features in all feature maps across all stages. For the $k$-th stage, we would like to find a mapping $S_k$ which selects and groups features in $U$ into a set of $n'$ representative features $\tilde{U}_k$ by

$$\tilde{U}_k = S_k U, \tag{3}$$

where $S_k \in \mathbb{R}^{n' \times n}$ and $\tilde{U}_k \in \mathbb{R}^{n' \times c}$. That is, we assemble $n'$ representative features from $n$ pixel-level features by their linear combinations. The map $S_k$ is the linear transformation which performs the linear dimensionality reduction by taking weighted averages over all pixel-level features.

[1] We refer the feature associated with a cell of the feature map as a pixel-level feature. Notice that a "pixel" of the feature map actually occupies a local patch in the input image.

We write the map $S_k$ as the product of two learnable maps, $C$ and $M_k$:

$$S_k = CM_k, \tag{4}$$

where $C \in \mathbb{R}^{n' \times m}$, $M_k \in \mathbb{R}^{m \times n}$ and $m$ is a parameter. The map $M_k$ is specific for the $k$-th stage while the map $C$ is shared across all stages. The maps $M_k$ and $C$ are formed as follows:

$$M_k = \sigma(f_M(A_k)), \tag{5}$$

$$C = \sigma(f_C(A)), \tag{6}$$

where $\sigma$ is the sigmoid function; $f_M$ and $f_C$ are two different functions defined by fully-connected layers; and $A = [A_1, A_2, \ldots, A_K]$ is the concatenation of all attention maps. Both $f_M$ and $f_C$ are parts of the end-to-end trainable FSA-Net and they are discovered through learning from training data. The use of a separable map for $S_k$ not only reduces the number of parameters, but also stabilizes the training. Furthermore, $L_1$ normalization is performed for each row of $S_k$ for more stable training.

Each row of the map $M_k$ can be folded into $K$ maps of the size $w \times h$, each of which represents how the pixel-level features spatially contribute to the representative feature corresponding to the specific row. Thus, each row of $M_k$ can be taken as a fine-grained structure that is salient to pose estimation. Figure 5 visualizes some maps.

Finally, we concatenate all $\tilde{U}_k$ together to form the final set of representative features, $\tilde{U} = [\tilde{U}_1, \tilde{U}_2, \ldots, \tilde{U}_K]$, where $\tilde{U} \in \mathbb{R}^{(n' \cdot K) \times c}$. The set of representative features $\tilde{U}_k$ is then fed into a feature aggregation method for obtaining the final set of features, $V \in \mathbb{R}^{K \times c'}$, for stage-wise regression.

### 3.6. Details of the architecture

Similar to the DeepCD [44] and SSR-Net, the FSA-Net has two streams. They are built with two basic building blocks $\mathrm{B_R}$ and $\mathrm{B_T}$ as:

$\mathrm{B_R}(\mathrm{c}) \equiv \{\text{SepConv2D}(3\times3, \text{c})\text{-BN-ReLU}\}$,

$\mathrm{B_T}(\mathrm{c}) \equiv \{\text{SepConv2D}(3\times3, \text{c})\text{-BN-Tanh}\}$,

where SepConv2D is separable 2D convolution; BN denotes batch normalization and c is a parameter. The structure of the first stream is $\{\mathrm{B_R}(16)\text{-AvgPool}(2\times2)\text{-}\mathrm{B_R}(32)\text{-}\mathrm{B_R}(32)\text{-AvgPool}(2\times2)\}$ - $\{\mathrm{B_R}(64)\text{-}\mathrm{B_R}(64)\text{-AvgPool}(2\times2)\}$ - $\{\mathrm{B_R}(128)\text{-}\mathrm{B_R}(128)\}$. The layers between each pair of brackets form a stage. The feature map at the end of a stage is the output of the stage. The structure of the second stream is $\{\mathrm{B_T}(16)\text{-MaxPool}(2\times2)\text{-}\mathrm{B_T}(32)\text{-}\mathrm{B_T}(32)\text{-MaxPool}(2\times2)\}$ - $\{\mathrm{B_T}(64)\text{-}\mathrm{B_T}(64)\text{-MaxPool}(2\times2)\}$ - $\{\mathrm{B_T}(128)\text{-}\mathrm{B_T}(128)\}$. Since there are three stages, the parameter $K$ is equal to 3 in our FSA-Net.

As for other parameters, in our current implementation, we set $w = 8$, $h = 8$ and $c = 64$ for the feature maps. We set $m = 5$ and $n' = 7$ for the fine-grained structure mapping, and $c' = 16$ for the feature aggregation module throughout all experiments.

Figure 3. Examples from the datasets. The first row is from the *300W-LP* synthetic dataset. In this dataset, the images at different poses are rendered, instead of being taken in the real world. The second row is from the *AFLW2000* dataset which contains many different real-world backgrounds and light conditions. The third row is from the *BIWI* dataset which was collected under the controlled environment.

## 4. Experiments

This section describes implementation, the datasets for training and testing, evaluation protocols, results, comparisons with other methods, and the ablation study.

### 4.1. Implementation

We used Keras with Tensorflow backend for implementing the proposed FSA-Net. For data augmentation in training, we applied random cropping and random scaling $(0.8 \sim 1.2)$ to training images. We used 90 epochs to train the network with the Adam optimizer with the initial learning rate $0.001$. The learning rate was reduced by a factor of $0.1$ every 30 epochs. The experiments were performed on a computer with an Intel i7 CPU and an GTX1080Ti GPU. The inference time of our model is around 1ms per image.

### 4.2. Datasets and evaluation protocols

**Datasets.** Three popular datasets for head pose estimation were adopted in the experiments: the *300W-LP* [48], *AFLW2000* [48], and *BIWI* [13] datasets, The *300W-LP* dataset [48] was derived from the 300W dataset [37] which unifies several datasets for face alignment with 68 landmarks. Zhu *et al*. used face profiling with 3D image meshing to generate $61,225$ samples across large poses and further expanded to $122,450$ samples with flipping [48]. The synthesized dataset is named as the 300W across Large Poses (*300W-LP*). The *AFLW2000* dataset [48] provides ground-truth 3D faces and the corresponding $68$ landmarks for the first $2,000$ images of the AFLW dataset [21]. The faces in the dataset have large pose variations with various illumination conditions and expressions. The *BIWI* dataset [13] contains $24$ videos of $20$ subjects in the controlled environment. There are a total of roughly $15,000$

frames in the dataset. In addition to RGB frames, the dataset also provides the depth image for each frame. Figure 3 shows examples from these three datasets. For training and evaluation on these datasets, we follow the following two common protocols.

**Protocol 1.** For this protocol, we follow the setting of Hopenet [35] whose goal is also landmark-free head pose estimation: training on the synthetic *300W-LP* dataset while testing on the two real-world datasets, the *AFLW2000* and *BIWI* datasets. Notice that, the same as the setting of Hopenet, when evaluating on the *BIWI* dataset, we do not use tracking and only considers samples whose rotation angles are within the range of $[-99°, +99°]$ with MTCNN [47] face detection. We compare several state-of-the-art landmark-based pose estimation methods using this protocol. The batch size we used for this protocol is 16.

**Protocol 2.** In this protocol, we used 70% of videos (16 videos) in the *BIWI* dataset for training, and the others (8 videos) for testing. The faces in the *BIWI* dataset are detected by MTCNN with the empirical tracking technique to avoid failure of face detection. Notice that this protocol was adopted by several pose estimation methods with different modalities such as RGB, depth, and time while our method only utilizes a single RGB frame. We used the batch size 8 for training in this protocol.

### 4.3. Competing methods

We compare our method with the following state-of-the-art methods for pose estimation. The first group of methods is landmark-based. **KEPLER** [22] predicts facial keypoints and pose at the same time with a modified GoogLeNet architecture. The coarse pose supervision is used for improving landmark detection. **FAN** [3] is a state-of-the-art landmark detection method. It is robust against occlusions and head poses. The method acquires multi-scale information by merging block features multiple times across layers. **Dlib** [20] is a standard face library which contains landmark detection, face detection, and several other techniques. **3DDFA** [48] uses CNNs to fit a 3D model to an RGB image. The dense 3D model allows alignment of the landmarks even with occlusions. **Hopenet** [35] is a landmark-free regression method. It employs ResNet and trains it using both MSE and cross-entropy losses.

There are also some head pose estimation methods which utilize multiple modalities. **VGG16** (RGB) and **VGG16+RNN** (RGB+Time) were proposed by Gu *et al*. [16]. They analyzed multiple possibilities of combining the CNN and RNN based on analysis of Bayesian filters. **Martin** [25] estimates head pose from depth images from a consumer depth camera by building and registering a 3D head model. **DeepHeadPose** [28] focuses on low-resolution RGB-D images. It uses both classification and regression to predict the estimation confidence.

| | MB | Yaw | Pitch | Roll | MAE |
|---|---|---|---|---|---|
| Dlib (68 points) [20] | - | 23.1 | 13.6 | 10.5 | 15.8 |
| FAN (12 points) [3] | 183 | 6.36 | 12.3 | 8.71 | 9.12 |
| Landmarks [35] | - | 5.92 | 11.86 | 8.27 | 8.65 |
| 3DDFA [48] | - | 5.40 | 8.53 | 8.25 | 7.39 |
| Hopenet ($\alpha=2$) [35] | 95.9 | 6.47 | 6.56 | 5.44 | 6.16 |
| Hopenet ($\alpha=1$) [35] | 95.9 | 6.92 | 6.64 | 5.67 | 6.41 |
| SSR-Net-MD [45] | **1.1** | 5.14 | 7.09 | 5.89 | 6.01 |
| FSA-Caps (w/o) | 2.9 | 5.27 | 6.71 | 5.28 | 5.75 |
| FSA-Caps (1×1) | **1.1** | 4.82 | 6.19 | 4.76 | 5.25 |
| FSA-Caps (var.) | **1.1** | 4.96 | 6.34 | 4.78 | 5.36 |
| FSA-Caps-Fusion | 5.1 | **4.50** | **6.08** | **4.64** | **5.07** |

Table 1. Comparisons with the state-of-the-art methods on the AFLW2000 dataset. All are trained on the 300W-LP dataset.

| | MB | Yaw | Pitch | Roll | MAE |
|---|---|---|---|---|---|
| 3DDFA [48] | - | 36.2 | 12.3 | 8.78 | 19.1 |
| KEPLER [22] | - | 8.80 | 17.3 | 16.2 | 13.9 |
| Dlib (68 points) [20] | - | 16.8 | 13.8 | 6.19 | 12.2 |
| FAN (12 points) [3] | 183 | 8.53 | 7.48 | 7.63 | 7.89 |
| Hopenet ($\alpha=2$) [35] | 95.9 | 5.17 | 6.98 | 3.39 | 5.18 |
| Hopenet ($\alpha=1$) [35] | 95.9 | 4.81 | 6.61 | 3.27 | 4.90 |
| SSR-Net-MD [45] | **1.1** | 4.49 | 6.31 | 3.61 | 4.65 |
| FSA-Caps (w/o) | 2.9 | 4.56 | 5.15 | 2.94 | 4.22 |
| FSA-Caps (1×1) | **1.1** | 4.78 | 6.24 | 3.31 | 4.31 |
| FSA-Caps (var.) | **1.1** | 4.56 | 5.21 | 3.07 | 4.28 |
| FSA-Caps-Fusion | 5.1 | **4.27** | **4.96** | **2.76** | **4.00** |

Table 2. Comparisons with the state-of-the-art methods on the BIWI dataset. All are trained on the 300W-LP dataset.

### 4.4. Results with protocol 1

In this scenario, pose estimation methods are trained with the 300W-LP dataset. Table 1 and Table 2 compare our FSA-Net with the state-of-the-art methods on the AFLW2000 and BIWI datasets, respectively. The mean absolute error (MAE) is used as the evaluation metric. In this protocol, the characteristics of the training and testing datasets are quite different. The training dataset is synthetic while the testing datasets are real. The landmark-free approaches can better accommodate the domain discrepancy between training and testing. Thus, the landmark-free methods (Hopenet, SSR-Net-MD and FSA-Net) perform better than landmark-based ones on both AFLW2000 and BIWI datasets. Figure 4 compares our model with Hopenet by showing a few examples. Both SSR-Net-MD and FSA-Net are more compact than Hopenet. All FSA-Net variants perform better than SSR-Net-MD. FSA-Caps denotes FSA-Net that uses capsule [36] for feature aggregation. There are three options for the scoring function: w/o for not applying fine-grained feature mapping, $1\times1$ for $1\times1$ convolution and

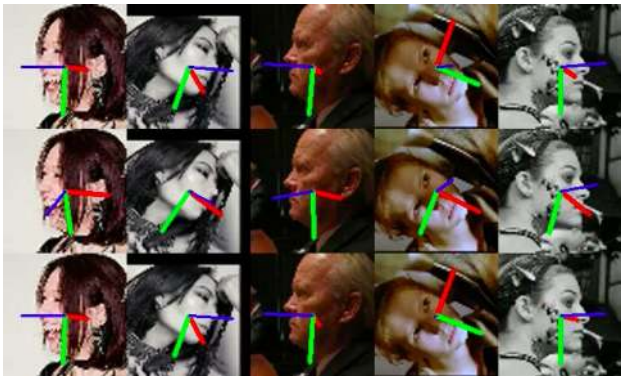


Figure 4. Pose estimation on the AFLW2000 dataset (protocol 1). From top to bottom, they are ground truth, results of Hopenet and our results. The blue line indicates the direction the subject is facing; the green line for the downward direction while the red one for the side. Best view in color.

var. for variance. Although $1\times1$ convolution has the potential for learning a better mapping from data, it could suffer from overfitting. Thus, from the experiments, it does not always lead to the best performance. We found that their fusion with simple average, denoted as FSA-Caps-Fusion, produces the most robust results. KEPLER [22] also intends to find the structure relation between keypoints, but our scheme of learning fine-grained structure mapping is much more effective than their iterative method.

### 4.5. Results with protocol 2

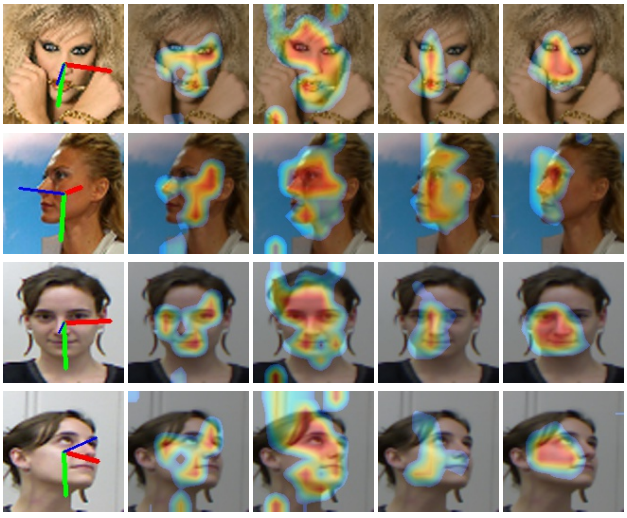
The BIWI dataset contains information from multiple modalities. Other than using color information within a single frame, it is possible to leverage depth or temporal information for improving performance. Table 3 reports performance of methods using different modalities. The RGB-based group only uses a single RGB frame while RGB+Depth and RGB+Time respectively utilize depth and temporal information in addition to color information. Our method (FSA-Caps-Fusion) only uses a single RGB frame and outperforms all other methods in its peer group. VGG16 is close but its model size is much bigger. Our model does not perform as well as methods using multi-modality information, but not too far from them. In addition, our method is the best on predicting the yaw angle, even outperforming those with multi-modality information.

### 4.6. Visualization

Figure 5 visualizes the fine-grained structures captured by our method. The model is the FSA-Caps ($1\times1$) model trained on the 300W-LP dataset. The first column shows the estimated poses. The rests are visualizations for how some representative features are aggregated from pixel-level features, one column for one feature. The heatmaps are the reshaped versions of the row vectors of $M_k$ recovered in Section 3.5. For example, the heatmaps in Figure 5(c) show that the forehead and the region of eyes are aggregated for the specific feature. The detected regions are similar across images but slightly different due to the head poses. As another example, Figure 5(e) focuses on the right cheek.

| Method | MB | Yaw | Pitch | Roll | MAE |
|---|---|---|---|---|---|
| **RGB-based** | | | | | |
| DeepHeadPose [28] | - | 5.67 | 5.18 | - | - |
| SSR-Net-MD [45] | **1.1** | 4.24 | 4.35 | 4.19 | 4.26 |
| VGG16 [16] | 500 | 3.91 | 4.03 | 3.03 | 3.66 |
| FSA-Caps-Fusion | 5.1 | **2.89** | 4.29 | 3.60 | 3.60 |
| **RGB+Depth** | | | | | |
| DeepHeadPose [28] | - | 5.32 | 4.76 | - | - |
| Martin [25] | - | 3.6 | **2.5** | **2.6** | 2.9 |
| **RGB+Time** | | | | | |
| VGG16+RNN [16] | >500 | 3.14 | 3.48 | **2.60** | **3.07** |

Table 3. Comparisons with the-state-of-art methods on the BIWI dataset. 70% of videos are used for training (16 videos) and 30% for testing (8 videos). There are three groups of methods that use information from different modalities: RGB-based, RGB+Depth and RGB+Time.



(a) (b) (c) (d) (e)

Figure 5. Visualizations of the discovered fine-grained spatial structures. The model is the FSA-Caps (1×1) trained on the 300W-LP dataset. The first column shows the estimated head poses. The other four columns display four spatial structures by heatmaps which visualize the folded versions of some rows of $M_k$ discovered by the model. They show how pixels are aggregated for a specific representative feature. The examples of the first two rows are from the AFLW2000 dataset, and those of the last two rows come from the BIWI dataset.

### 4.7. Ablation Study

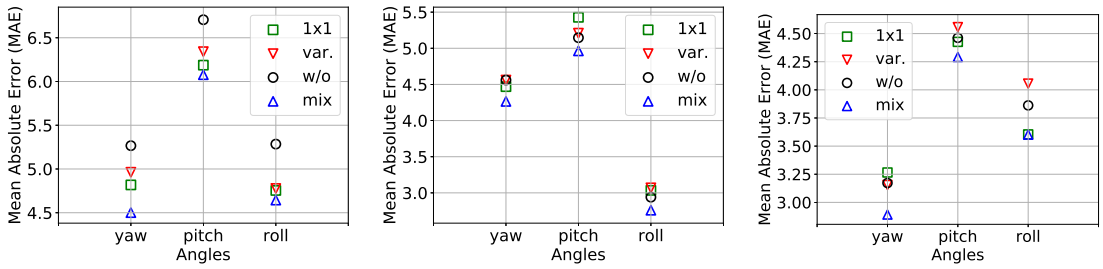
We have conducted the ablation study for understanding the influence of individual components, including different

| testing set | AFLW2000 (protocol 1) | | | | | | | BIWI (protocol 1) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| method | SSR | FSA-Net | | | | | | SSR | FSA-Net | | | | | |
| aggregation | - | - | | | Capsule [36] | | | - | - | | | Capsule [36] | | |
| pixelwise scoring | - | w/o | 1×1 | var. | w/o | 1×1 | var. | - | w/o | 1×1 | var. | w/o | 1×1 | var. |
| model size (MB) | 1.1 | 0.5 | 0.8 | 0.8 | 2.9 | 1.1 | 1.1 | 1.1 | 0.5 | 0.8 | 0.8 | 2.9 | 1.1 | 1.1 |
| MAE | 6.01 | 5.54 | 5.48 | 5.41 | 5.75 | 5.25 | 5.36 | 4.65 | 4.61 | 4.53 | 4.16 | 4.22 | 4.31 | 4.28 |
| MAE (late fusion) | - | 5.14 | | | **5.07** | | | - | 4.19 | | | **4.00** | | |

Table 4. Ablation study for different aggregation methods (no aggregation and Capsule) and the different pixelwise scoring functions for protocol 1. The results are the MAEs of the yaw, pitch, and roll angles. SSR denotes SSR-Net-MD [45].

| testing set | BIWI (protocol 2) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| method | SSR-Net-MD [45] | FSA-Net | | | | | | | | |
| aggregation | - | - | | | Capsule [36] | | | NetVLAD [1] | | |
| pixelwise scoring | - | w/o | 1×1 | var. | w/o | 1×1 | var. | w/o | 1×1 | var. |
| model size (MB) | 1.1 | 0.5 | 0.8 | 0.8 | 2.9 | 1.1 | 1.1 | 0.6 | 0.8 | 0.8 |
| MAE | 4.26 | 3.95 | 4.01 | 3.83 | 3.84 | 3.77 | 3.92 | 3.97 | 3.88 | 3.88 |
| MAE (late fusion) | - | 3.75 | | | **3.60** | | | 3.68 | | |

Table 5. Ablation study over different aggregation methods (no aggregation, Capsule and NetVLAD) and the different pixelwise scoring functions under protocol 2. The results are the MAEs of the yaw, pitch, and roll angles.




(a) AFLW2000 (protocol 1) (b) BIWI (protocol 1) (c) BIWI (protocol 2)

Figure 6. Comparisons over each angle for different testing datasets and corresponding protocols. We divide the components of FSA-Caps-Fusion into three parts, $1 \times 1$, var., and w/o variants. The legend "mix" represents the fusion model.

aggregation methods (none, capsule, NetVLAD), and different pixelwise scoring functions (none, $1 \times 1$ convlution, or variance). Table 4 and Table 5 report the results. Since our method is based on SSR-Net-MD, its performance is also listed as a reference. The results are improved by using capsule or NetVLAD as the feature aggregation. This means that state-of-the-art aggregation methods can be naturally combined with our method. Figure 6 shows detailed comparison over the yaw, pitch and roll angles for several settings. While a single scoring function model does not always achieve good results, the fusion ensemble model guarantees the best result in every case, showing that complementary information is learned in different model variants.

## 5. Conclusion

In this paper, we propose a new way to acquire more meaningful aggregated features with the fine-grained spatial structures. By defining learnable and non-learnable scoring functions of the pixel-level features, we are able to learn complementary model variants. Experiments show that the ensemble of these variants outperforms the state-of-the-art methods (both landmark-based and landmark-free ones) while its model size is around $100\times$ smaller than those of previous methods. Furthermore, its estimation on the yaw angle is even more accurate than those methods with multi-modality information such as the RGB-D or RGB-Time recurrent model. We show that it is possible to improve regression results by learning meaningful intermediate features. Although we only demonstrate on the pose estimation problem, we believe that the idea can be extended to other regression problems as well.

**Acknowledgement** This work was supported in part by Ministry of Science and Technology (MOST) under grants 107-2628-E-001-005-MY3 and 108-2634-F-007-009, and MOST Joint Research Center for AI Technology and All Vista Healthcare under grant 108-2634-F-002-004.

## References


[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5297–5307, 2016.

[2] Relja Arandjelovic and Andrew Zisserman. All about VLAD. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[3] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3D facial landmarks). In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017.

[4] Kaidi Cao, Yu Rong, Cheng Li, Xiaoou Tang, and Chen Change Loy. Pose-robust face recognition via deep residual equivariant mapping. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[5] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face alignment by explicit shape regression. *International Journal of Computer Vision (IJCV)*, 107(2):177–190, 2014.

[6] Feng Ju Chang, Anh Tuan Tran, Tal Hassner, Iacopo Masi, Ram Nevatia, and Gerard Medioni. FacePoseNet: Making a case for landmark-free face alignment. In *Proceedings of Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, 2017.

[7] Dong Chen, Shaoqing Ren, Yichen Wei, Xudong Cao, and Jian Sun. Joint cascade face detection and alignment. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2014.

[8] Eunji Chong, Nataniel Ruiz, Yongxin Wang, Yun Zhang, Agata Rozga, and James M. Rehg. Connecting gaze, scene, and attention: Generalized attention estimation via joint modeling of gaze and scene saliency. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018.

[9] Grigorios G. Chrysos, Epameinondas Antonakos, Patrick Snape, Akshay Asthana, and Stefanos Zafeiriou. A comprehensive performance evaluation of deformable face tracking "in-the-wild". *International Journal of Computer Vision (IJCV)*, 126(2-4):198–232, 2018.

[10] Timothy F. Cootes, Christopher J. Taylor, David H. Cooper, and Jim Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[11] Daniel F. DeMenthon and Larry S. Davis. Model-based object pose in 25 lines of code. In *Proceedings of European Conference on Computer Vision (ECCV)*, 1992.

[12] Piotr Dollár, Peter Welinder, and Pietro Perona. Cascaded pose regression. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[13] Gabriele Fanelli, Matthias Dantone, Juergen Gall, Andrea Fossati, and Luc Van Gool. Random forests for real time 3D face analysis. *International Journal of Computer Vision (IJCV)*, 101(3):437–458, 2013.

[14] Gabriele Fanelli, Thibaut Weise, Juergen Gall, and Luc Van Gool. Real time head pose estimation from consumer depth cameras. In *Joint Pattern Recognition Symposium*, pages 101–110. Springer, 2011.

[15] Rohit Girdhar and Deva Ramanan. Attentional pooling for action recognition. In *Proceedings of Neural Information Processing Systems Conference (NIPS)*, 2017.

[16] Jinwei Gu, Xiaodong Yang, Shalini De Mello, and Jan Kautz. Dynamic facial analysis: From Bayesian filtering to recurrent neural network. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[17] Kuang-Jui Hsu, Yen-Yu Lin, and Yung-Yu Chuan. Deepco$^3$: Deep instance co-segmentation by co-peak search and co-saliency detection. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[18] Kuang-Jui Hsu, Chung-Chi Tsai, Yen-Yu Lin, Xiaoning Qian, and Yung-Yu Chuang. Unsupervised CNN-based co-saliency detection with graphical optimization. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018.

[19] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G. Schwing. Unsupervised video object segmentation using motion saliency-guided spatio-temporal propagation. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018.

[20] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[21] Martin Koestinger, Paul Wohlhart, Peter M. Roth, and Horst Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *Proceedings of International Conference on Computer Vision Woskshops*, 2011.

[22] Amit Kumar, Azadeh Alavi, and Rama Chellappa. KEPLER: Keypoint and pose estimation of unconstrained faces by learning efficient H-CNN regressors. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, 2017.

[23] Donghoon Lee, Hyunsin Park, and Chang D. Yoo. Face alignment using cascade Gaussian process regression trees. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[24] Lin Liang, Rong Xiao, Fang Wen, and Jian Sun. Face alignment via component-based discriminative search. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 72–85. Springer, 2008.

[25] Manuel Martin, Florian Van De Camp, and Rainer Stiefelhagen. Real time head model creation and head pose estimation on consumer depth cameras. In *Proceedings of The 2nd International Conference on 3D Vision (3DV)*, 2014.

[26] Iain Matthews and Simon Baker. Active appearance models revisited. *International Journal of Computer Vision (IJCV)*, 60(2):135–164, 2004.

[27] Gregory P. Meyer, Shalini Gupta, Iuri Frosio, Dikpal Reddy, and Jan Kautz. Robust model-based 3D head pose estimation. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[28] Sankha S Mukherjee and Neil Martin Robertson. Deep head pose: Gaze-direction estimation in multimodal video. *IEEE Transactions on Multimedia (TMM)*, 17(11):2094–2107, 2015.

[29] Erik Murphy-Chutorian and Mohan Manubhai Trivedi. Head pose estimation in computer vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(4):607–626, 2009.

[30] Duy-Kien Nguyen and Takayuki Okatani. Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[31] Rajeev Ranjan, Vishal M. Patel, and Rama Chellappa. HyperFace: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.

[32] Rajeev Ranjan, Swami Sankaranarayanan, Carlos D. Castillo, and Rama Chellappa. An all-in-one convolutional neural network for face analysis. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, 2017.

[33] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision (IJCV)*, 126(2-4):144–157, 2016.

[34] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2011.

[35] Nataniel Ruiz, Eunji Chong, and James M. Rehg. Fine-grained head pose estimation without keypoints. In *Proceedings of International Conference on Computer Vision Woskshops*, 2017.

[36] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. In *Proceedings of Neural Information Processing Systems Conference (NIPS)*, pages 3856–3866, 2017.

[37] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *Proceedings of International Conference on Computer Vision Woskshops*, 2013.

[38] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep convolutional network cascade for facial point detection. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[39] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning GAN for pose-invariant face recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[40] Chung-Chi Tsai, Weizhi Li, Kuang-Jui Hsu, Xiaoning Qian, and Yen-Yu Lin. Image co-saliency detection and co-segmentation via progressive joint optimization. *IEEE Transactions on Image Processing (TIP)*, 2019.

[41] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: Convolutional block attention module. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018.

[42] Xuehan Xiong and Fernando De la Torre. Supervised descent method and its applications to face alignment. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[43] Xuehan Xiong and Fernando De la Torre. Global supervised descent method. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2664–2673, 2015.

[44] Tsun-Yi Yang, Jo-Han Hsu, Yen-Yu Lin, and Yung-Yu Chuang. Deepcd: Learning deep complementary descriptors for patch representations. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017.

[45] Tsun-Yi Yang, Yi-Hsuan Huang, Yen-Yu Lin, Pi-Cheng Hsiu, and Yung-Yu Chuang. SSR-Net: A compact soft stage-wise regression network for age estimation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

[46] Feifei Zhang, Tianzhu Zhang, Qirong Mao, and Changsheng Xu. Joint pose and expression modeling for facial expression recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[47] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.

[48] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z. Li. Face alignment across large poses: A 3D solution. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 146–155, 2016.

[49] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.