Leveraging Convolutional Pose Machines for Fast and Accurate Head Pose Estimation

Yuanzhouhan Cao[1], Olivier Canévet [1] and Jean-Marc Odobez[1,2]

***Abstract*— We propose a head pose estimation framework that leverages on a recent keypoint detection model. More specifically, we apply the convolutional pose machines (CPMs) to input images, extract different types of facial keypoint features capturing appearance information and keypoint relationships, and train multilayer perceptrons (MLPs) and convolutional neural networks (CNNs) for head pose estimation. The benefit of leveraging on the CPMs (which we apply anyway for other purposes like tracking) is that we can design highly efficient models for practical usage. We evaluate our approach on the Annotated Facial Landmarks in the Wild (AFLW) dataset and achieve competitive results with the state-of-the-art.**

I. Introduction

Head pose estimation has been a difficult research topic in computer vision for decades. It can be exploited for head gesture recognition [4], and more importantly, as a proxy for gaze [5], it is an important non-verbal cue that can inform about the attention of people by itself [1]. As such, it can be used in many human analysis tasks and in particular for human-robot interaction (HRI) [7], [18], [8], social event analysis [12], driver assistance system [2], and gaze estimation [6]. The purpose of head pose estimation is to predict the head pose expressed as three rotation angles (roll, pitch, yaw). Conventional methods estimate the head pose either by fitting facial points to a 3D model [21], [13], [19], matching facial point clouds with pose candidates through a triangular surface patch descriptor [14], or using bayesian methods, especially for tracking tasks [17].

Recently, head pose estimation considerably benefitted from the success of convolutional neural networks (CNNs) [9]. Ranjan et al. [16] propose a unified deep learning framework for head pose estimation, face detection, landmark localization and gender recognition. Patacchiola et al. [15] propose a deep learning model to estimate the head pose of in-the-wild face images. Other works also include depth data in CNNs, like Borghi et al. [2] which generate face from depth data to predict the head pose of drivers.

Apart from head pose, body pose and body landmark detection have also considerably improved, e.g. with the introduction of the convolutional pose machines (CPMs) [3]. The CPMs aim at localizing the body landmarks (eyes, ears, nose, shoulders, etc.) and the body limbs (arms, legs, etc.) and leverage the context by iteratively refining its predictions. Fig. 1 depicts CPM results on some face images.

[1] Idiap Research Institute, Switzerland. yuanzhouhan.cao@idiap.ch, olivier.canevet@idiap.ch, odobez@idiap.ch

[2] Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland.

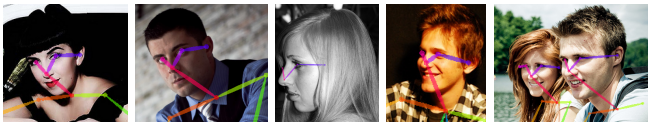

Fig. 1: Example of body landmark detection with CPM [3]. We are interested in using the predictions of the nose, eyes, and ears to estimate the head pose (yaw, pitch, and roll)

In this paper, we propose to use the landmark detections as well as the features of the CPMs to predict the head pose. Our work is motivated by the fact that the detections of the CPMs (namely the nose, eyes, and ears) are extremely reliable (see Fig. 1), and that the CPMs is often applied anyway for scene perception, e.g. for tracking people. In this context, the head pose can be obtained as a by-product on top of the CPMs.

Our head pose estimation system is illustrated in Fig. 2. For a given input color image, we first apply the CPMs to extract features including keypoints, confidence maps, and feature maps. Then we train a predictor to output the head pose from them. In this contex, our contributions are:

- We investigate several strategies to leverage the output of the CPMs (landmarks, confidence maps, features);
- We investigate several predictors (multilayer perceptrons, convolutional neural networks);
- We investigate several strategies for head pose estimation (pose angle regression, pose likelihood regression).
- We achieve competitive head pose predictors making an error of less than 10° for roll, pitch, and yaw.

Our system can be viewed as a simple strategy on top of the CPMs to estimate the head pose of detected persons[1].

II. Convolutional pose machines

The first part of our head pose estimation approach is to extract features capturing appearance information and facial keypoint relationships. We apply the convolutional pose machines (CPMs) [3] to extract the features.

The CPM is a real-time multi-person keypoint detection model (see Fig. 1 for an illustration). In the rest of the paper, we equally use keypoint and landmark to name the body part predicted by the CPMs. We illustrate the architecture of the CPMs in Fig. 2. It takes as input a color image of size $h \times w$ and simultaneously output body part confidence maps and affinity fields that encode part-to-part association. The color image is first fed into a VGG-19 network, generating a set of feature maps $\mathbf{F}$ that is input to the following stages.

[1]The code to train the model and to run a real time demo on a webcam is available at `https://gitlab.idiap.ch/software/openheadpose`.

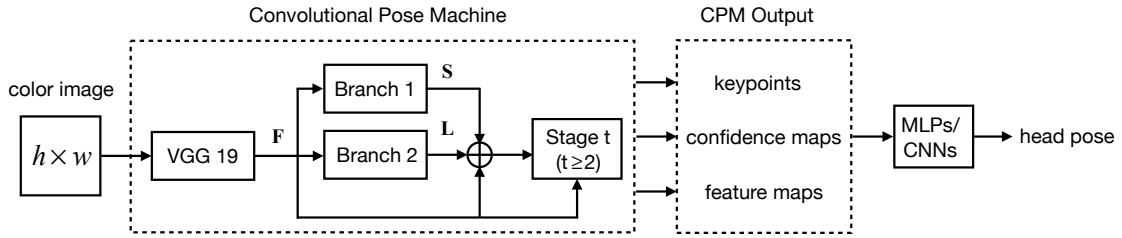

Fig. 2: Architecture of our head pose estimation system. It takes as input the color images and outputs CPM features. The CPM features are fed into MLPs or CNNs for head pose estimation.

Each stage contains two branches that have same network structures. The top branch produces a set of confidence maps **S** and the bottom branch produces a set of affinity fields **L**. After each branch, the feature maps **F**, the confidence maps **S** and the affinity fields **L** are concatenated to be the input of the following stage. More stages lead to more refined predictions. In this paper, we use the CPMs with 6 stages.

What makes the CPM very efficient is that it makes use of the context and of powerful VGG features, by refining its own predictions all along stages $t \geq 2$, which leads to accurate localization of the body joints.

During training, two L_2 loss functions are applied at the end of each stage, one at each branch respectively. The loss values of both branches are added up and backpropagated to update network weights. In our head pose estimation, we apply the CPMs trained on the Microsoft COCO dataset [10] with 18 body keypoints (including nose, eyes, ears, and shoulders) to extract the features. We directly apply this network without finetuning.

III. Proposed method

In this section, we elaborate our head pose estimation approach. Specifically, for an input image, we first obtain the keypoints, the confidence maps, and the feature maps through the CPMs. Then we train multilayer perceptrons (MLPs) or convolutional neural networks (CNNs) to predict the head pose expressed as angles of pitch, roll and yaw.

A. Keypoint-based head pose estimation

The outputs of the CPM are confidence maps and part affinity fields. The position of a body keypoint can be obtained by selecting the pixel with the maximum confidence value that is above a predefined threshold. If all the values in a confidence map are smaller than the threshold, then the corresponding keypoint is considered not to be in the input image. The position of a keypoint is represented as a two dimensional coordinate vector (x, y). Since our task is to estimate the head pose, we only consider 8 keypoints in the upper body: nose, neck, eyes, ears and shoulders. The obtained coordinate is normalized as:

$$(x_n, y_n) = \left(\frac{x - x_c}{w}, \frac{y - y_c}{h} \right), \quad (1)$$

where (x_c, y_c) is the center of the face region and (w, h) is the size of the face region. In practice, we use the center of nose and eyes:

$$(x_c, y_c) = \left(\frac{x_{nose} + x_{leye} + x_{reye}}{3}, \frac{y_{nose} + y_{leye} + y_{reye}}{3} \right) \quad (2)$$

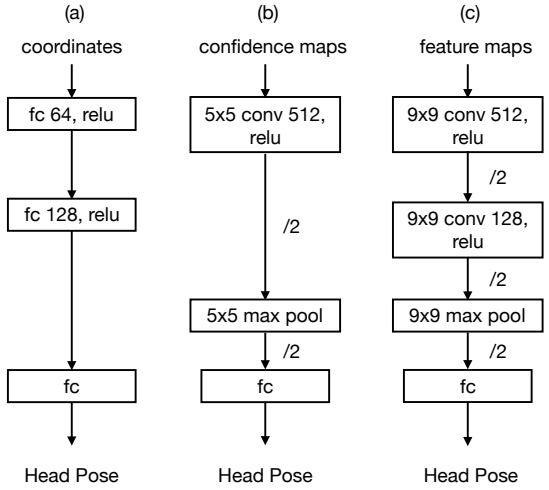

Fig. 3: Structures of our proposed head pose estimation models. (a) MLPs taking as input keypoint coordinates. (b) CNNs taking as input the confidence maps of keypoints. (c) CNNs taking as input feature maps. The number of hidden nodes in the last fully-connected layer depends on the loss.

to be the face center. The width w of the face region is defined as the horizotal distance between two eyes, and the height h is defined as the larger vertical distance between the nose and the two eyes. The normalized coordinate vectors of all the keypoints are concatenated to be the input of our head pose estimation model. Since some keypoints may not exist in an input image, the concatenated coordinate vector contains missing values. We apply the probabilistic principal analysis (PPCA) [20] to fill these missing values.

We train multilayer perceptrons (MLPs) for head pose estimation. The structure of our MLPs is illustrated in Fig. 3(a). It contains 3 fully-connected layers and rectified linear units (ReLUs) are applied after the first two fully-connected layers. The MLPs take as input the normalized coordinates of keypoints and output the head pose.

B. CPM-feature based head pose estimation

In order to achieve more accurate head pose estimation, for an input image, we extract deeper features from different layers of the CPMs and train convolutional neural networks (CNNs). Specifically, we extract the confidence maps and the feature maps. The structure of CNNs trained on the confidence maps are illustrated in Fig. 3(b). It contains one convolution layer, one ReLU, one max pooling layer and one fully-connected layer.

As for the feature maps, we extract the output of two different layers from the CPMs: the output of the VGG-19 network (**F** in Fig. 2), and the output of the second last convolution layer of the Branch 2 in the last stage. The dimension of the two sets of feature maps is 128. The structure of the CNNs trained on these feature maps is illustrated in Fig. 3(c). It contains two convolution layers followed by two ReLUs, one max pooling layer and one fully-connected layer.

Our CNNs for head pose estimation take as input the face regions in the CPM features. We crop the face regions based on the positions of nose and eyes as illustrated in Fig. 4. We first get the vertical distance d between the nose and the center of the eyes. Then we crop a square of size $5d \times 5d$ around the nose, with $3d$ above the nose, $2d$ below the nose, $2.5d$ left and right to the nose. The positions of nose and eyes are predicted by the CPMs. In our experiments, we also crop the face regions according to the ground-truth face rectangles of the dataset. The height and the width of the cropped face regions are rescaled to 128.

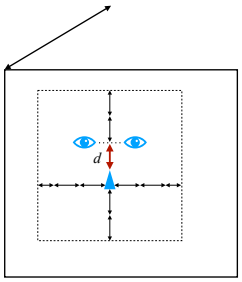

Fig. 4: Illustration of cropping. We crop the face regions in the confidence maps and the feature maps based on the positions of nose and eyes.

C. Loss function

Our head pose estimation models are trained by minimizing the mean squared error (MSE) between the predictions and the ground-truth values, defined as:

$$L(\mathbf{y},\mathbf{y}^*) = \frac{1}{3}\frac{1}{b}\sum_{p=1}^{3}\sum_{i=1}^{b}(y_p(i) - y_p^*(i))^2. \quad (3)$$

We consider two types of model predictions: angles and angle likelihoods. When our models directly predict the head pose expressed as roll, pitch and yaw angles, the last fully-connected layers in Fig. 3 have 3 hidden nodes. The value of b is 1 in Eq. 3 and $\mathbf{y} = (a_r, a_p, a_y)$ and $\mathbf{y}^* = (a_r^*, a_p^*, a_y^*)$ are model prediction and ground-truth respectively.

The angles of roll, pitch and yaw are periodic. Consider a ground-truth value of 0°, the predictions of 1° and 359° should have the same loss values. Directly predicting the pose angles does not consider this discontinuity. In addition, the predictions lack confidence. In order to solve these drawbacks, we convert the ground-truths from angles to angle likelihoods as illustrated in Fig 5. Specifically, we first evenly discretize the range of $[-\pi, \pi]$ into several bins. Then for a ground-truth angle d in radians, we generate two gaussian distributions with a variance σ, and the mean values are d and $d - 2\pi$ (shown in red and blue) respectively. For each bin, we assign the maximum value of the two gaussian distributions to be the likelihood of the corresponding angle range.

When our models predict angle likelihoods, b in Eq. 3 is the number of bins. The last fully-connected layers in Fig. 3 have $3b$ hidden nodes. $\mathbf{y} = (\mathbf{l_r}^\top, \mathbf{l_p}^\top, \mathbf{l_y}^\top)$ and $\mathbf{y}^* = (\mathbf{l_r^*}^\top, \mathbf{l_p^*}^\top, \mathbf{l_y^*}^\top)$ in Eq. 3 are predicted and ground-truth likelihoods respectively.

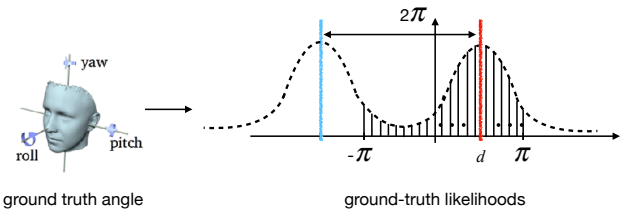

Fig. 5: Conversion of ground-truth angles to likelihoods. The values of likelihoods are generated from two gaussion distributions with mean values of d and $d - 2\pi$ (shown in red and blue) respectively, where d is the ground-truth angle in radians.

IV. Experiments

We now present the results of our method of head pose estimation. We show that our method is able to leverage the CPMs output and yields competitive results with the state-of-the-art Hyperface model.

A. Dataset

We evaluate our head pose estimation on the Annotated Facial Landmarks in the Wild (AFLW) dataset [11]. The AFLW is a large-scale, multi-view, real-world face dataset gathered from Flickr, exhibiting a large variety in face appearance (e.g. pose, expression, ethnicity, age, gender) as well as general imaging and environmental conditions. It contains about 25k faces annotated with facial landmarks, face rectangles, head pose, etc. In our experiments, the input to the convolutional pose machine (CPMs) network is the AFLW faces with the largest possible contexts.

B. Evaluation protocol

We divide the AFLW into train, validation, and test sets: for all the faces in the AFLW dataset, there are 17,081 faces with nose and two eyes annotated, and 19,887 faces with nose and two eyes detected by the CPMs. We select an intersection of 15,230 faces, and randomly split these images into a training set of 9,138 images, and a validation set and a test set of 3,046 images each.

To allow a fair comparison with the Hyperface method which also uses the AFLW dataset, we contact the authors to get their own train and test sets. This other "split" of the AFLW is only used in Table V for comparison.

We apply the following measures to evaluate the head pose estimation:

- mean absolute error (MAE): $\frac{1}{N}\sum_n |y_n - y_n^*|$, where y_n^* is the ground truth value (roll, pitch or yaw) of image n, y_n is the estimated value by our method, and N is the total number of images in the test set. This MAE represents the error that our method does in estimating the three angles.
- accuracy with threshold τ: percentage of y_n s.t. $|y_n - y_n^*| = \delta < \tau$, where y_n is the estimated value and y_n^* the ground truth. The "accuracy below a threshold" is an interesting cue because an error of 10° (for example) in the estimation is actually very small to the human eye. This score considers correct an estimation which is below threshold τ from the correct value.

C. Keypoint-based head pose estimation results

In this section, we evaluate our MLPs trained on the coordinate vectors of keypoints (see Sec. III-A). As explained in Sec. IV-A, we only kept the images from AFLW for which the CPMs could detect the nose and both eyes.

We train two MLPs, one by using 3 keypoint coordinates (nose and two eyes), and another one by using 5 keypoints (two ears in addition). These keypoints are the coordinates of the detected keypoints by the CPMs (see Fig. 1 as an illustration).

Table I shows the results of the two networks. The MLPs trained on 3 keypoints achieve an error of 10.65° on average for the yaw, and an accuracy of 62.5% with a 10° threshold. The MLPs trained on 5 keypoints achieve an error of 9.19° for the yaw, and an accuracy of 70.9% with a 10° threshold. We can see that using more keypoints yields better performance.

As a sanity check, we also train MLPs with the ground-truth annotations of the AFLW dataset. This corresponds to the ideal case where we train with perfect locations (as opposed to keypoints detected by CPMs as in the previous 2 cases). We see in Table I that with more accurate locations of face landmarks, we can achieve more satisfactory pose estimations: 5.46° average error for the yaw, and 96.5% accuracy with a 10° threshold.

TABLE I: Head pose estimation results based on facial keypoint positions. The first two rows are MLPs trained with 3 keypoints and 5 keypoints estimated by the CPMs. The last row are the results of MLPs trained with 5 ground-truth keypoints.

	Accuracy (%)			Error
	$\delta < 5^\circ$	$\delta < 10^\circ$	$\delta < 15^\circ$	MAE
3 keypoints from CPM	r: 68.8 p: 33.6 y: 35.5	r: 87.9 p: 60.1 y: 62.5	r: 93.5 p: 78.7 y: 77.5	r: 5.04 p: 9.53 y: 10.65
5 keypoints from CPM	r: 71.9 p: 43.4 y: 44.9	r: 89.2 p: 72.6 y: 70.9	r: 94.1 p: 89.0 y: 83.6	r: 4.72 p: 7.52 y: 9.19
5 AFLW annotations (sanity check)	r: 85.8 p: 49.9 y: 55.6	r: 97.9 p: 78.6 y: 85.5	r: 99.4 p: 92.1 y: 96.5	r: 2.77 p: 6.45 y: 5.46

D. CPM-feature based head pose estimation results

In this section we present our head pose estimation results based on the confidence maps and the feature maps respectively. We also show the results of the angle likelihood regression.

Confidence map-based. Each confidence map represents the probability of existence of a body landmark (or keypoint) at all the positions of an input image. We train convolutional neural networks with different number of keypoints and show the results in the first 4 rows of Table II. Specifically, the 3 keypoints are nose and eyes, the 5 keypoints are the 3 keypoints plus ears, and the 8 keypoints are the 5 keypoints plus neck and shoulders. All the confidence maps are extracted from the last stage of the CPM network (stage 6). The face regions in the confidence maps are cropped using the ground-truth face rectangles without contexts. In order to crop the neck and shoulders in the confidence maps with 8 keypoints, we crop the ground-truth face rectangles with some contexts. Specifically, 25% of the hight above, 75% of the hight below, 50% of the width left and right. From Table II we can see that experiment with more keypoint confidence maps leads to better performance (row "5 confidence maps from stage 6" better than row "3 confidence maps from stage 6"). However, using 8 keypoints does not further improve the performance, this is caused by the inconsistency of shoulders and head, as a front looking face may have multiple shoulder positions.

The confidence maps in the aforementioned experiments are extracted from the last stage (stage 6) of the CPM. We have also investigated the use of features in the earlier stages of the CPM, namely stage 1 and stage 3 (rows "5 confidence maps from stage 1" and "5 confidence maps from stage 3"), which are less refined that in stage 6. We observe that that our CNN models trained on the confidence maps of stage 6 have the best performance. We can conclude that the late feature are better for head pose estimation, which was expected as the late features are more refined, and better localize the body landmarks.

We have also investigated the cropping of the face regions from confidence maps. So far, the face regions are cropped using the ground-truth face rectangles in the AFLW dataset. However, in practice, the ground-truth rectangles are not available. We crop the face areas using the CPM estimated facial keypoints as illustrated in Sec. III-B. We also crop the face areas with some additional context. Specifically, we crop a square of size $10d \times 10d$ centred at the nose, where d is the vertical distance between the nose and the center of eyes in Fig. 4. From the table, rows "5 confidence maps + estimated cropping" and "[...] context", we can see that the results of cropping using ground-truth face rectangles (i.e. AFLW annotations) outperform the results of cropping using estimated facial keypoints. The performance can be further improved with more accurate face localization. Some cropping examples are shown in Fig. 6.

Fig. 6: Cropping examples. The first two rows are the examples of cropping using estimated facial keypoints without and with context respectively. The second two rows are the examples of cropping using ground-truth face rectangles without and with context respectively.

Overall, we can see from Table II that using 5 keypoint confidence maps yields the best estimation of the head pose.

Feature map-based. We are now interested in whether the features of the first part of the CPM (i.e. VGG features) are sufficient for head pose estimation. To this purpose, we train a first network taking as input feature maps from the VGG-19 network (**F** in Fig. 2), and another network that takes as input the second last convolution layer of Branch 2 in the last stage. Both of the feature maps have 128 channels. We crop the face regions using the ground-truth face regions without context. The results are illustrated in Table III, from which we can see that the results using the feature maps of the last stage significantly outperform the results using the VGG-19 feature maps. This makes sense since the features of stage 6 are specifically trained for body parts, especially nose, ears, and eyes.

Angle likelihood regression. In all the aforementioned experiments, the outputs of the CNNs are pose angles. The CNNs are trained by minimizing the mean square errors between ground-truth and estimated angles. In this section, we train CNNs by minimizing the mean square errors between ground-truth and estimated angle likelihoods, as illustrated in Sec. III-C. The inputs of the CNNs are confidence maps of 5 facial keypoints generated from the last stage of the CPM, and the face regions are cropped with the ground-truth face rectangles without context. We discretize the range of $[-\pi, \pi]$ into different number of bins, and use the Gaussian distributions with different variance σ. We report the errors in Table IV. As we can see from the table that the best performance is achieved when the number of bins is 180 and $\sigma = 0.3$. But the results are roughly the same as by directly regressing the angles.

E. Stage-of-the-art comparison

In this section, we compare our head pose estimation method with the Hyperface method [16].

TABLE II: Head pose estimation based on confidence maps with the variation of number of keypoints. By default, the confidence maps are extracted from the last stage. The "context" indicates that the confidence maps are cropped with contexts, and the "estimated" indicates that the confidence maps are cropped based on the CPM estimated keypoints.

	Accuracy (%)			Error
	$\delta < 5^\circ$	$\delta < 10^\circ$	$\delta < 15^\circ$	MAE
3 confidence maps from stage 6	r: 80.2 p: 44.7 y: 44.1	r: 95.8 p: 76.6 y: 74.7	r: 98.2 p: 91.8 y: 89.2	r: 3.38 p: 6.91 y: 7.40
8 confidence maps from stage 6	r: 79.1 p: 45.0 y: 45.2	r: 94.7 p: 77.8 y: 74.2	r: 97.8 p: 92.4 y: 89.2	r: 3.59 p: 6.69 y: 7.45
8 confidence maps from stage 6 + context	r: 73.2 p: 43.0 y: 45.0	r: 92.7 p: 75.0 y: 73.6	r: 97.2 p: 90.4 y: 87.4	r: 4.06 p: 7.13 y: 7.64
5 confidence maps from stage 6	r: 80.3 p: 46.5 y: 46.9	r: 95.7 p:79.0 y:76.2	r: 98.2 p: 92.9 y: 90.1	r: 3.37 p: 6.55 y: 7.04
5 confidence maps from stage 1	r: 76.0 p: 44.9 y: 42.7	r: 93.1 p: 76.9 y: 72.4	r: 97.0 p: 92.0 y: 86.6	r: 3.91 p: 6.76 y: 8.02
5 confidence maps from stage 3	r: 73.8 p: 42.4 y: 44.5	r: 92.3 p: 73.6 y: 73.2	r: 97.3 p: 89.9 y: 87.5	r: 4.13 p: 7.25 y: 7.66
5 confidence maps + estimated cropping	r: 75.5 p: 43.9 y: 40.8	r: 92.3 p: 74.6 y: 68.3	r: 96.0 p: 90.5 y: 82.7	r: 4.09 p: 7.12 y: 9.08
5 confidence maps, + estimated cropping + context	r: 66.3 p: 42.0 y: 37.3	r: 87.4 p: 72.5 y: 64.6	r: 94.3 p: 89.3 y: 79.2	r: 5.15 p: 7.47 y: 9.96

TABLE III: Head pose estimation using feature maps extracted from different layers of the CPM network. The first row are the results of using the output of the VGG-19 network (**F** in Fig. 2), the second row are the results of using the output of the second last convolution layer of Branch 2 in the last stage.

	Accuracy (%)			Error
	$\delta < 5^\circ$	$\delta < 10^\circ$	$\delta < 15^\circ$	MAE
Features **F** from VGG	r: 43.6 p: 31.7 y: 20.1	r: 67.2 p: 58.3 y: 35.9	r: 80.7 p: 77.1 y: 48.2	r: 9.14 p: 9.87 y: 18.60
Last features from stage 6	r: 79.9 p: 49.0 y: 51.4	r: 95.8 p:81.8 y: 80.5	r: 98.3 p: 94.4 y: 92.8	r: 3.44 p: 6.10 y: 6.41

TABLE IV: Head pose estimation using angle likelihood regression. The ground-truth likelihoods are generated with different number of discretizing bins and σ in Sec. III-C.

	$\sigma = 0.1$	$\sigma = 0.3$	$\sigma = 0.5$
180 bins	r: 5.33 p: 7.36 y: 7.47	r: 3.86 p: 7.14 y: 7.04	r: 3.88 p: 7.09 y: 7.38
240 bins	r: 5.75 p: 7.48 y: 7.65	r: 4.07 p: 7.28 y: 7.25	r: 4.05 p: 7.28 y: 7.52
360 bins	r: 5.87 p: 7.72 y: 7.85	r: 4.18 p: 7.32 y: 7.34	r: 4.19 p: 7.36 y: 7.71

TABLE V: Head pose estimation on the Hyperface split of the AFLW dataset. The first 4 rows are results of our approach and the last row are the results in [16]. Our models are trained using feature maps. The "GT" and "Est" indicate the face region cropping scheme, and the "angle" and "likelihood" indicate the loss function during training.

	Accuracy (%)			Error
	$\delta < 5°$	$\delta < 10°$	$\delta < 15°$	MAE
GT, angle	r: 71.4 p: 50.4 y: 45.6	r: 91.1 p: 79.6 y: 72.3	r: 96.2 p: 93.1 y: 86.1	r: 4.58 p: 6.34 y: 8.21
GT, likelihood	r: 73.5 p: 48.9 y: 46.5	r: 92.1 p: 79.1 y: 73.0	r: 96.6 p: 91.6 y: 86.4	r: 4.21 p: 6.57 y: 8.43
Est, angle	r: 64.8 p: 46.5 y: 39.1	r: 85.6 p: 76.5 y: 65.1	r: 93.6 p: 90.0 y: 76.8	r: 5.54 p: 7.23 y: 11.38
Est, likelihood	r: 66.6 p: 42.9 y: 40.2	r: 86.3 p: 74.0 y: 67.5	r: 93.6 p: 88.4 y: 81.2	r: 5.37 p: 7.57 y: 10.58
Hyperface	r: 76.0 p: 51.0 y: 46.0	r: 95.0 p:81.0 y: 76.0	r: 97.0 p: 95.0 y: 89.0	r: 3.92 p: 6.13 y: 7.61

For a fair comparison with the Hyperface method which also uses the AFLW, we contact the authors to get their own train and test sets of the AFLW. So here, neither the train set nor the test set is the same as in the previous sections.

Our CNNs are trained on the feature maps that are extracted from the second last convolution layer of the Branch 2 in the last stage of the CPM. During training, we apply the regression on both pose angles and likelihoods. The face regions in the features maps are cropped using both ground-truth face rectangles as well as the CPM estimated facial keypoints without contexts. We show the results in Table V. Note that the accuracies in [16] are reported as curves in a 2-D coordinate system, with the x axis to be the thresholds and y axis to be the accuracies. We obtain the rough accuracies of $\delta < 5°, \delta < 10°, \delta < 15°$ from these curves. We can see that our approach, yields competitive results with the Hyperface. More importantly, our method takes much less time to process. As reported in [16], on a GTX TITAN-X GPU, it takes the Hyperface 3 seconds output head pose from an image. We test our method on a GTX 1080-TI GPU, and it takes around 0.0001 seconds to output head pose from a confidence map, and around 0.2 seconds to output head pose from an image (including the CPM processing)

V. Conclusion

We have proposed a head pose estimation approach leveraging the convolutional pose machines. Our method takes as input either facial keypoints, or confidence maps, or feature maps of the CPMs. We have shown that such method can obtain competitive performance with the state-of-the-art.

Given the impact of face localization on performance, one future direction is to learn how the face should be cropped before applying our pose estimation method. Overall, with the success of the CPM, our system can be seen as a simple method to get head pose as a by-product of the CPM.

VI. Acknowledgement

This work was supported by the European Union under the EU Horizon 2020 Research and Innovation Action MuMMER (MultiModal Mall Entertainment Robot), grant agreement no. 688147, `http://mummer-project.eu/`.

References

[1] S. Ba and J. Odobez. A study on visual focus of attention recognition from head pose in a meeting room. In *Proc. Workshop on Machine Learning for Multimodal Interaction (MLMI)*, May 2006.

[2] G. Borghi, M. Venturelli, R. Vezzani, and R. Cucchiara. Poseidon: Face-from-depth for driver pose estimation. 2017.

[3] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[4] C. Chen, Y. Yu, and J.-M. Odobez. Head nod detection from a full 3d model. In *Int. Conf. on Computer Vision Workshop,*, 2015.

[5] K. Funes and J.-M. Odobez. Person independent 3d gaze estimation from remote rgb-d cameras. In *ICIP*, 2013.

[6] K. A. Funes Mora and J.-M. Odobez. Gaze estimation in the 3d space using rgb-d sensors: Towards head-pose and user invariance. *Int. J. Comp. Vis.*, 2017.

[7] M. A. Goodrich and A. C. Schultz. Human-robot interaction: A survey. *Found. Trends Hum.-Comput. Interact.*, 1(3), 2007.

[8] M. F. Jung. Affective grounding in human-robot interaction. In *IEEE Int. Conf. Hum.-Robot Interact.*, 2017.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*. 2012.

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Proc. Eur. Conf. Comp. Vis.*, 2014.

[11] P. M. R. Martin Koestinger, Paul Wohlhart and H. Bischof. Annotated Facial Landmarks in the Wild: A Large-scale, Real-world Database for Facial Landmark Localization. In *Proc. First IEEE Int. Workshop. Benchmarking Facial Image Anal. Technol.*, 2011.

[12] S. Muralidhar, L. S. Nguyen, D. Frauendorfer, J.-M. Odobez, M. Schmid Mast, and D. Gatica-Perez. Training on the job: Behavioral analysis of job interviews in hospitality. In *ACM ICMI*, 2016.

[13] G. P. Meyer, S. Gupta, I. Frosio, D. Reddy, and J. Kautz. Robust model-based 3d head pose estimation. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2015.

[14] C. Papazov, T. K. Marks, and M. Jones. Real-time 3d head pose and facial landmark estimation from depth images using triangular surface patch features. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2015.

[15] M. Patacchiola and A. Cangelosi. Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods. *Pattern Recogn.*, 71, 2017.

[16] 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. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2017.

[17] E. Ricci and J. Odobez. Learning large margin likelihood for realtime head pose tracking. In *IEEE International Conference on Image Processing*, november 2009.

[18] S. Sheikhi and J. Odobez. Combining dynamic head pose and gaze mapping with the robot conversational state or attention recognition in human-robot interactions. *Pat. Recog. Letters*, 66:81–90, Nov. 2015.

[19] M. Storer, M. Urschler, and H. Bischof. 3d-mam: 3d morphable appearance model for efficient fine head pose estimation from still images. In *Proc. IEEE Int. Conf. Comp. Vis. Workshops*, 2009.

[20] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *J. of the Royal Stat. Society*, 61:611–622, 1999.

[21] Y. Yu, K. A. F. Mora, and J. Odobez. Robust and accurate 3d head pose estimation through 3dmm and online head model reconstruction. In *Int. Conf. Auto. Face Gesture Recogn.*, 2017.