Using a deformation field model for localizing faces and facial points under weak supervision

Marco Pedersoli[†] Tinne Tuytelaars[†] Luc Van Gool [†‡]

[†] KU Leuven, ESAT/PSI - iMinds
firstname.lastname@esat.kuleuven.be

[‡] ETH Zürich, CVL/D-ITET
vangool@vision.ee.ethz.ch

Abstract

Face detection and facial points localization are interconnected tasks. Recently it has been shown that solving these two tasks jointly with a mixture of trees of parts (MTP) leads to state-of-the-art results. However, MTP, as most other methods for facial point localization proposed so far, requires a complete annotation of the training data at facial point level. This is used to predefine the structure of the trees and to place the parts correctly. In this work we extend the mixtures from trees to more general loopy graphs. In this way we can learn in a weakly supervised manner (using only the face location and orientation) a powerful deformable detector that implicitly aligns its parts to the detected face in the image. By attaching some reference points to the correct parts of our detector we can then localize the facial points. In terms of detection our method clearly outperforms the state-of-the-art, even if competing with methods that use facial point annotations during training. Additionally, without any facial point annotation at the level of individual training images, our method can localize facial points with an accuracy similar to fully supervised approaches.

1. Introduction

Even if the problem of detecting faces seems practically solved, this is true only when considering *well aligned frontal faces*. In the general case, the problem is still challenging because perspective deformations due to the different poses/orientations of the face/camera generate a huge space of variability which can only be covered by a huge amount of samples. At the same time, interest is rapidly moving to facial analysis which requires the precise localization of facial points. In terms of annotation, this typically requires a much more time consuming procedure, where for each example each facial point needs to be annotated.

Recent methods for object detection [7] have shown that aligning the object can often produce better detection accuracy or similar detection accuracy but with a reduced set of training data. In particular, for faces, Zhu *et al*. [33] have shown that using a mixture of trees of parts (MTP) connected with spring-like deformation costs can lead, with a limited number of samples, to performance comparable to commercial face detectors generally trained with millions of samples [22]. Furthermore, as parts are placed on facial landmarks, the same model can be used for facial point localization as well as pose estimation.

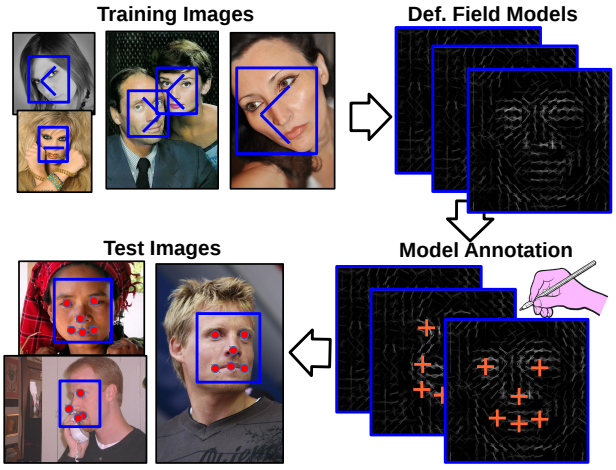


Figure 1. **Overview of our method.** Knowing only the location and orientation of faces at training time suffices to learn a model that can not only detect faces, but also localize their facial points.

In this work, we extend that approach. Instead of modeling faces as trees of parts, we model them as a densely and uniformly distributed set of parts connected with pairwise connections, forming a graph. The immediate advantage of this representation compared to the tree of parts is that the model does not need to know where the facial points are, because parts are placed uniformly over the entire face. The aligned locations of the parts are estimated during learning as latent variables.

This deformation model has loops and therefore its optimization is in general too expensive for detection. However, the inference procedure proposed in [18] makes the detection computationally feasible for up to 100 object parts. This model is well suited for perspective-like deformations and is general enough to automatically and jointly learn the face appearance, align the faces and learn the pairwise deformation costs without any facial point annotation. With this approach we show improved detection capabilities with less supervision (and just using few images). Additionally, we show that the generated alignment is good enough to be used for unsupervised facial point localization. In practice, without any additional learning we can manually select any number of facial points on the model representation (Fig. 1) and then use the model to localize the facial points on a new face.

The paper is structured as follows. In section 2 we discuss how our work relates to previous work. Then in section 3 we define our deformation model, how to learn it and how to localize facial points. Finally, section 4 reports on experiments comparing our model with the state of the art, while in section 5 we draw conclusions.

2. Related work

Face detection has been broadly studied. Here, due to lack of space, we limit ourselves to methods that are highly connected to ours. For a complete review on face detection and facial points localization we refer to recent surveys [9, 30, 31].

Viola and Jones [26] introduced the first detector able to correctly detect most of the faces in an image with a limited number of false positives. However, when considering faces in unconstrained environments, with any possible orientation, shadow, and occlusion, the real performance of the method is still quite low. To detect faces with different orientations, Chang *et al*. [10] learn different models for different discrete orientations and then at test time use a hierarchical cascade structure for a fast selection of the correct model. This improves detection, but needs a large amount of training data, because each model needs to be trained with samples at the correct orientation. In contrast, our model can adapt to the local and global deformations of a face. Therefore, the same sample is recycled for different face orientations producing a reduced computational cost, a higher number of samples per model and thus better performance.

A classical approach for facial point localization is Active Appearance Models (AAMs) [2, 17, 14] and more recently Constrained Local Models (CLM) [19]. Unfortunately those models, to work properly, need a good initialization, otherwise they get stuck in poor local minima. Instead, in our case, as we jointly perform face detection and landmark localization, there is no need for an initialization procedure. Other recent approaches are based on boosted regressors [25] and conditional regression forests [4]. They have shown promising results, but they need a large amount of annotated training images.

Our approach is similar to elastic graph matching methods [13, 16, 27], where the facial landmarks are connected in a deformable graph. However, in our model we do not need the annotated facial points, because a dense grid of points is placed around the face and then, during training the best deformation for these points is learned.

In terms of features, our approach is similar to deformable part models (DPM) [7], because we use a dense scan of parts over HOG features [3] at multiple scales. Everingham *et al*. [6] use a pictorial structure model to build a facial point detector, while in [24] a DPM is trained with structured output support vector machines to directly optimize the localization of the facial points. However, in contrast to previous methods, in our model parts are not limited to be connected with a star or tree structure. Instead, they form a graph.

Our work was inspired by the work of Zhu *et al*. [33] and its extension [28], where it is shown that a mixture of trees of parts placed on annotated facial points obtains state-of-the-art results for face detection, pose estimation and facial point localization. Here we show that this way of tackling the problem has further unexplored potential. By changing the tree mixtures for graph mixtures (that can still be optimized in a reasonable time) we can automatically learn the structure of the faces and align them without the need for annotated facial points. Another recent approach for face detection and facial point estimation is based on learning exemplars and use them to detect faces and transfer their annotations [21]. Again, the number of training faces needed to obtain good performance is quite high.

Unsupervised alignment is also a well studied topic [11, 15, 23]. Congealing, initially proposed by Learned-Miller [15] and then further extended and improved [11], enforces data alignment by finding the warping parameters that minimize a cost function on the ensemble of training images. Tong *et al*. [23] applied the algorithm for localizing facial points with a limited number of annotated images, while Zhu *et al*. [32] applied a similar alignment on a dense deformable map. However, these techniques do not consider negative examples and expect the face to be already coarsely aligned *i.e*. previously detected. Instead we show that joining face detection and facial points localization is beneficial for both tasks. As far as we know, this is the first work performing unsupervised facial points localization as a side effect of deformable face detection. In our case the face deformation is a latent variable and we implicitly align the training images to the detector model because the alignment minimizes the recognition loss.

3. Model

We build our model on the deformation field model (DFM) proposed for object detection in [18]. In this section we first revise the DFM and then we show how to adapt it to detecting faces. Given an image I and a set of learned weights w_m, we define the score generated by the mixture m of our model as a combination of appearance and deformation:

$$S(I, L, H_m, w_m) = A(I, L, w_m) - D(L, H_m, w_m). \quad (1)$$

where $L = \{l_i : i \in \mathcal{P}\}$ with $l_i = (l_i^x, l_i^y)$ representing the location of part i and $H_m = \{h_i : i \in \mathcal{P}\}$ with $h_i = (h_i^x, h_i^y)$ representing the anchor point of part i. The score of the model appearance is produced by a set of parts $\mathcal{P}$ placed on the image I:

$$A(I, L, w_m) = \sum_{i \in \mathcal{P}} \left\langle w_{m,i}^A, \Phi^A(I, l_i) \right\rangle, \quad (2)$$

where $w_{m,i}^A$ are the weights for the mixture m associated to the features Φ^A (*e.g.* HOG features) extracted at location l_i. The deformation cost penalizes relative displacement of connected parts:

$$D(L, H_m, w_m) = \sum_{ij \in \mathcal{E}} \left\langle w_{m,ij}^D, \Phi^D(l_i - h_i, l_j - h_j)] \right\rangle. \quad (3)$$

$\mathcal{E}$ are the edges of the graph connecting neighboring parts and $w_{m,ij}^D$ are the weights for the mixture m associated to the deformation features $\Phi^D(d_i, d_j) = (|d_i^x - d_j^x|, |d_i^y - d_j^y|, (d_i^x - d_j^x)^2, (d_i^y - d_j^y)^2)$.

As we use a 4-connected grid model, the corresponding graph has loops and standard dynamic programming optimization cannot be used. Instead, we consider the problem as an instance of a CRF optimization where nodes are the object parts, node labels are the locations of the parts in an image and edges are the pairwise connections between parts. As proposed in [18] for each mixture and for each scale we globally maximize Eq.(1) using alpha expansion [1], which is fast. To detect multiple instances in the same image, we iteratively re-run the algorithm penalizing the locations of a previous detection. For loopy connected deformation models, this iterative optimization is much faster than a sliding window approach, but it provides similar accuracy. The computational cost of the algorithm is linear in the number of locations in the image and it is empirically linear also in the number of parts. For a complete analysis of speed and quality of this deformation model we refer to [18].

3.1. DFM for faces

To effectively use DFM we adapt the algorithm to the specific problem of detecting faces and facial points. In order to properly localize facial points we have to use a relatively high number of parts so that each point can be localized by a different part. At the same time, we want to detect small faces, therefore the global model should have a relatively low resolution. However, a low resolution model with many parts would not give good results in terms of facial point localization because coarse parts are not discriminative enough to properly localize the facial features. Thus, whereas in the original DFM parts are placed on a regular grid side by side, here, we introduce an overlap of 50% that allows for bigger parts without increasing the model resolution. More in detail, in all our experiments we use parts of 4×4 HOG cells with 2 cells overlap. For selecting the model resolution (and consequently the number of parts) we use two conditions: (i) maximum number of parts defined by the maximum computational cost which is linear in the number of parts, (ii) ensure that at least 85% of the training data can be represented with the given resolution.

Also, as faces have quite a rigid structure[1], to avoid unlikely configurations, we modify the deformation features as $\Phi^F(d_i, d_j) = (cl(d_i^x - d_j^x), cl(d_i^y - d_j^y), (d_i^x - d_j^x)^2, (d_i^y - d_j^y)^2)$. $cl()$ is a non-linear function defined as:

$$cl(d) = \begin{cases} +\infty & if \quad d < -\mu \\ d & otherwise \end{cases} \quad (4)$$

where μ is the size of a part. It forbids parts to cross over each other, thus enforcing more regularity in the deformation structure[2].

Finally, to select among the different mixtures and overlapping hypotheses we run a non-maximal-suppression that suppresses all the bounding boxes that overlap more than 30% with the highest scoring ones.

3.2. Learning

Given a set of positive and negative images, the bounding boxes $\mathcal{B}$ of the faces and their pose (yaw), we want to learn a vector of weights w^* such that:

$$\begin{aligned} w^* = \arg\min_w \{ & \frac{1}{2} \max_m |w_m|^2 \\ & + C \sum_{n=1}^{M} \sum_{k=1}^{K} \max(0, 1 + \max_{L,m} S(I_{n,k}, L, H_m, w_m)) \\ & + C \sum_{n=1}^{|\mathcal{B}|} \max(0, 1 - \max_{L,m} S(\mathcal{B}_n, L, H_m, w_m)) \}. \end{aligned} \quad (5)$$

This minimization is an instance of the latent SVM problem [7]. The locations of the object parts L and mixture

[1] in the sense that the topological location of the different parts is always the same, but their distance can change and this is why a deformation model is useful

[2] Although $cl(d)$ is not symmetric, it can still be optimized with alpha expansion as shown in [1]

m are the latent variables. C is the trade-off between loss and regularization. In all our experiments we fix C to 0.001. The regularization is maximized over mixtures m to enforce comparable scores for each mixture. For negative examples, as we are interested in ranking detections, we select the first K best detections generated from each of M negative images. For positive examples we collect the cropped region $B_n \in \mathcal{B}$ around each bounding box.

As opposed to binary SVMs, here the problem is not symmetric. Due to the maximization of the latent variables, the loss for the negative samples is convex, while the loss for the positive samples is concave. This is solved using an iterative procedure. Given an initial w we find the latent values L and m for the positive samples. Then, fixing those, we find a new w optimizing the convex problem.

In the ideal case, when we can optimally maximize the score of Eq. (1), the loss of the positive samples can only decrease at each iteration and, hence, the algorithm converges [29]. Unfortunately, the alpha expansion algorithm puts only a weak bound on the quality of the solution [1]. As suggested in [18], to keep the convergence, we maintain a buffer with the previously assigned values for the latent variables. When the new assignment is effectuated, we maintain it only if it produces a lower loss (higher score); otherwise the old assignment is restored.

The optimization is effectuated using stochastic gradient descent [20]. As the number of negative samples is exponential, to use a limited amount of memory we use negative mining as proposed in [7]. During learning, the weights associated to the deformation costs w_m^D are forced to be positive to avoid unwanted configurations. With stochastic gradient descent we can impose positiveness on the weights by just re-projecting, at each update of w, all negative weights to zero.

3.3. Initialization

As the problem defined in Eq. 5 is not convex, the final quality of the model highly depends on the quality of the initialization of the latent variables. Initially, to avoid wrong configurations, the latent variable values are restricted to fewer configurations. Then, slowly, the restrictions are relaxed and a refined model can be learned. More in detail:

- **Split into mixtures**: we split uniformly the range of yaw angles of the faces in the dataset based on the number of mixtures that we want to build. For roll and pitch we assume that those rotations can be accounted for in the deformation model and they do not need a separate mixture. For instance, for a 2-mixtures model, we split the yaw angle between 0-45 and between 45-90. We consider only the absolute value of the angles because examples facing left can be placed in the same mixture with examples facing right (see below). Then, for each mixture m we crop the corresponding bounding boxes $\mathcal{B}^m$ of the face and rescale them to a fixed scale.

- **Left-right alignment**: we align left and right facing examples to train a single mixture with more positive samples. Then at test time we run the inference with the learned model as well as with the horizontally flipped version, so that we can detect faces facing both sides. To this end, we define an alignment energy as:

$$\sum_{n=1}^{|\mathcal{B}^m|} |\Phi^A(\mathcal{B}_n^m, L^*) - \frac{1}{|\mathcal{B}^m|} \sum_{n=1}^{|\mathcal{B}^m|} \Phi^A(\mathcal{B}_n^m, L^*)|^2, \quad (6)$$

which measures the norm of the variance on each cell for all the samples of a given mixture. L^* is the resting-like configuration of the parts, when there is no deformation cost. We minimize this energy just selecting random samples and flipping them horizontally. If the energy with the flipped sample is lower than before, then the sample is kept flipped, otherwise, the old configuration is restored. We repeat this procedure for 10 times the number of samples in the mixture.

- **Initial appearance model**: with the samples separated by yaw angle and correctly aligned $\mathcal{B}^m$ for each mixture m and a set of random cropped regions $\mathcal{R}$ from images not containing faces we train a first appearance model based on standard SVM optimization with the latent variables fixed:

$$\begin{aligned} w_m^* = \arg\min_w \{ & \frac{1}{2}|w_m|^2 \\ & + C \sum_{n=1}^{|\mathcal{R}|} \max(0, 1 + \langle w_m^A, \Phi^A(\mathcal{R}_n, L^*) \rangle) \\ & + C \sum_{n=1}^{|\mathcal{B}^m|} \max(0, 1 - \langle w_m^A, \Phi^A(\mathcal{B}_n^m, L^*) \rangle \}. \end{aligned} \quad (7)$$

- **Initial deformation model**: we initialize the deformation weights connecting the part i and j as $w_{m,ij}^D = |w_{m,i}^A| + |w_{m,j}^A|$ so that they are comparable to the corresponding appearance weights. In most of the cases this initial configuration does not allow for any deformation. However, it allows global displacements of the mixture model to rigidly align the mixture to the face, as in our deformation model a global displacement is not penalized. Then, during training, the deformation weights are regularized and after some latent SVM iterations they will be small enough to allow for deformations.

- **Complete model**: we concatenate all the initial appearance weights w_m^A with the corresponding deformation weights w_m^D to form the complete w that is used

to initialize Eq. 5 and therefore start the latent SVM optimization.

- **Initialization without pose**: in the experiments we also tried to learn the model without using the facial pose. In this case we first perform the left-right alignment as previously explained on the entire training data. Afterwards, we perform k-means on the extracted features, with k equal to the number of desired mixtures. In this way each cluster now represents one mixture and the following steps of the initialization are the same as before.

3.4. Facial point localization

Once the training is completed we obtain a set of deformable templates representing the different views of a face. In Fig. 2 we show the positive weights associated to the HOG features learned for 3 different viewpoints. From this visualization we can easily recognize the face structure and therefore we can manually annotate the facial points that we want to localize in a new image.

Then, for each point we can find which part of the grid it belongs to and anchor it to the corresponding part. In this way, when applying the detector on a new image the facial point will follow the location of the part that it is anchored to. As during learning we trade-off appearance and deformation, also on test images we can expect that the parts will distort to adapt to the current image.

If a facial point is placed at the edge between two parts, its real placement on the image could vary a lot depending on which part we decide it is attached to. To avoid such problem each facial point is attached to the 4 closest parts. The final point location on the image is then the bi-linear interpolation of the location of the point on the four parts. This procedure reduces the quantization effect due to the fact that parts have a finite size. In practice in our experiments the interpolation always gives better results and we use it in all experiments.

3.5. Evaluating the Facial Alignment

The previous procedure is useful to estimate the location of facial points for real applications, like facial emotion recognition or person identification. However, it is based on a subjective localization of the facial points on the object model. A more direct way to estimate how good a model is in aligning faces is to re-project for each image the ground truth annotations onto the object model. If most of the points fire at the same location on the object model, then the alignment is well done. In this sense, for each facial point we evaluate the standard deviation of the re-projections of the annotated faces.

Again, the re-projection is computed using bi-linear interpolation. In practice for each annotated facial point, the 4 closest parts are detected and then the location of the point on the model is their weighted mean. An example of annotated facial points re-projection is shown in Fig. 2.

4. Experiments

4.1. Datasets

We train our method using 900 samples from two well known datasets: MultiPIE [8] and Labeled faces in-the-wild (LFW) [12]. We use 900 samples for training to have a fair comparison with previous methods [33] that used the same number of samples. In contrast to the other methods, we use only the location of the bounding box and the pose (yaw) of the face, but not the facial point locations. We use MultiPIE, that is a collection of images taken in a controlled environment, to perform an analysis of the model parameters. Afterwards, for a comparison with other state-of-the-art methods we train with LFW which contains unconstrained images and has a better representation of the "in-the-wild" data distribution. For both datasets, for collecting negative samples we use the negative images of the INRIA dataset [3] which do not contain faces. The test is effectuated on Annotated Faces in-the-wild (AFW) proposed in [33].

4.2. Number of Mixtures

In Table 1 we evaluate the effect of changing the number of mixtures of the model using a grid of 10×10 parts with part size of 4×4 HOG cells and 2 cells overlap. We train our model on MultiPIE considering 300 frontal views and 600 lateral views spanning from +-15 to +-90 degrees as in [33]. For each configuration we evaluate on AFW the detection average precision (AP) and the average standard deviation of the projection of the facial points on the model as explained in sec. 3.5. For the average precision, we consider a detection as correct if its overlap with the ground truth bounding box is more than 50%, as in [5]. We use the average standard deviation of the re-projection of the annotated facial points (as percentage of the face size) to estimate the capability of alignment of the model on the test samples. A high standard deviation means that the localization is poor, while a small one is an indicator that the model aligns well with the test images.

From the table we can see that increasing the number of mixtures (up to 8) leads to a better facial point estimation (lower standard deviation). However, increasing the number of mixtures reduces the number of samples per mixture and thus, with more than 8 mixtures the facial point localization becomes worse. For detection we can see that starting from 6 mixtures already gives near optimal performance. As the computational time is linear in the number of mixtures, for the next experiments we select the configuration with 6 components which has a quite good AP and facial

mixtures	4	6	8	10	12
AP(%)	86.1	89.6	90.1	91.4	90.3
Std(% face)	3.02	1.28	1.08	1.12	1.15

Table 1. Detection (% AP) and facial point localization (average standard deviation of the re-projected facial points on the object model as explained in sec. 3.5) for different numbers of mixtures.

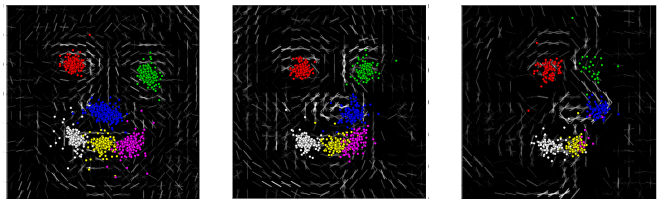


Figure 2. **Re-projection of the facial point annotations on LFW**. Each color represents the projection of a different facial point on the object model as spesified in sec. 3.5. In this figure as well as in the following ones, even though the model is composed of parts we do not show them for the sake of clarity.

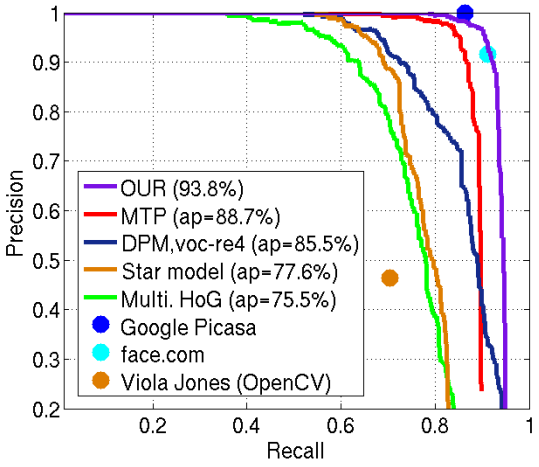


Figure 3. **Precision-recall curves for AFW**. With only 900 training images our method outperforms most of the other methods. Note that this evaluation is done on all the test faces of AFW.

point localization, but its computational cost is lower.

4.3. Comparison with other methods

For comparing with other methods we select a configuration with similar characteristics with MTP which is the most similar method to ours and the current state-of-the-art. We use a model with 10×10 parts each composed of 4×4 HOG cells with 50% overlap and 6 mixtures, which come from 3 different orientation models and their vertically flipped versions. On our machine our model takes around 100 seconds on a single core to detect faces and localize facial points on an image of 640×480 pixels. This is comparable with the MTP independent model of Zhu *et al*. [33] that, on our machine with a single core takes around 80 seconds. In contrast to [33], we do not need high resolution images for training, therefore we use 900 faces from LFW, which has a more varied distribution of samples and therefore avoids over-fitting.

Detection. In Fig. 3 we compare our approach with other methods on all the faces of AFW dataset. The results of other methods are provided by [33]. Our method obtains an average precision of 93.8 which is more than 5 points above MTP [33], that needs the facial points annotations to be trained. Compared with DPM which also does not need annotated facial points, our model outperforms it by more than 8 points. Notice that we use exactly 6 mixtures like DPM. Our better performance is due to the fact that we use a better deformation model which needs only 6 mixtures, while [33] uses 18. Using fewer mixtures allows each mixture to use more training samples and therefore to generalize better to any possible face.

Facial points localization. While in the previous subsection we evaluate the facial point localization in terms of standard deviation of the projected facial points on the object model, here, to compare results with other supervised methods, we manually define the location of the facial points on the object model based on visual inspection and then we use the method explained in sec. 3.4 to estimate the facial points on new images. In Fig. 6 (a) we compare our model with other methods. To compare with the results provided by [33], we evaluate on AFW's faces larger than 150 pixels in height, although our model can detect and localize facial points also on smaller ones. Our method, even if unsupervised, at an average error rate inferior to 5% of the face size can correctly estimate the facial points for 60.5% of the faces. This is superior to some well known methods such as Oxford [6], CML and multi. AAMs [19], even if those methods, not detecting the face, have been given the advantage to localize the facial points using the ground truth bounding box. We also compare our same model without allowing for deformation. In this case the performance at 5% of the face size drops to 25.8%. This shows that a good deformation model is really important for good facial point localization.

Compared with face.com [22] and MTP our method has a lower score. However, it does not need any facial point annotation on training images. Also, in the evaluation protocol, if one point is occluded and the method places it, the localization error is set to infinite and the entire facial points would be considered wrong. Considering that our method has a limited number of mixtures and does not reason about occlusions, that evaluation is slightly unfair. In Fig. 6 (b) we show the cumulative localization error of the entire face and for each part of the face, this time without penalizing the localization of occluded facial points. In this case the performance of our model rises to 75%. Considering the

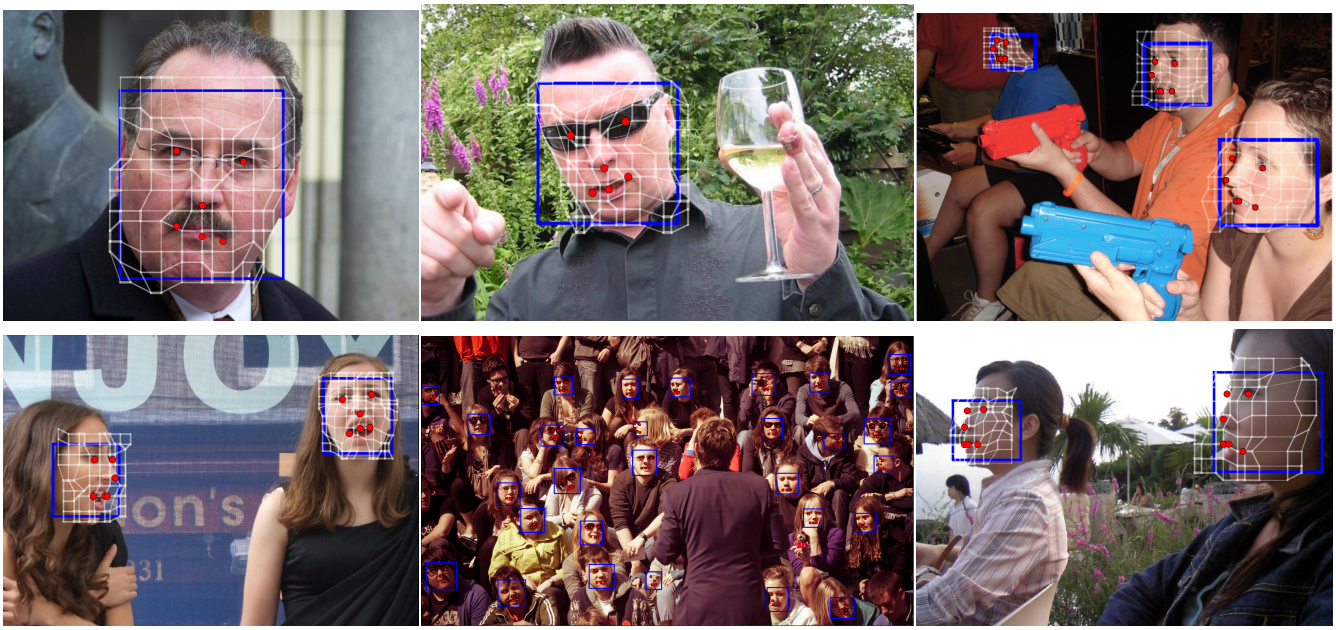


Figure 4. **Face detections and facial point localization in the wild.** We show the detected faces (blue boxes), the facial points estimation (red dots) and the distortion of the deformation field (white lines).

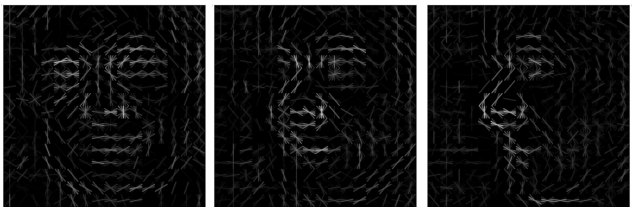
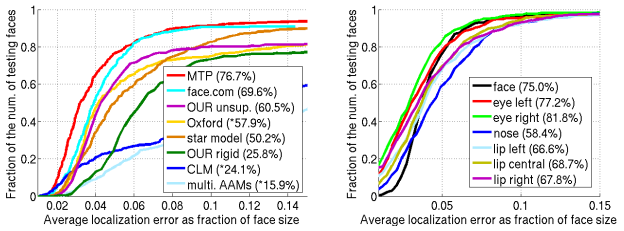


Figure 5. **Mixtures obtained without pose estimation**. Even without knowing the pose of the faces at training time we are able to learn meaningful models.



(a) (b)

Figure 6. **Cumulative error distribution for facial point localization on AFW.** (a) Comparison with other methods. (b) Cumulative error for each facial landmark.

evaluation of each part separately shows that eyes, being very discriminative, are the best localized points, while the nose is the worse one. In Fig. 4 we show some examples of detections and facial point estimation on AFW.

Unannotated pose. Finally, we test our method with even less annotation. Instead of using the annotation of the head pose for selecting which mixture every sample should be initially associated with, we use k-means on the HOG features to split the data. As shown in Fig. 5, the obtained mixtures are still quite clearly defined and it is still possible to distinguish the different poses of the head. With this approach we obtain an AP of 91.5% which is still higher than the other proposed approaches and a facial point estimation of 53.5% faces at an average error smaller than 5% the face size.

4.4. Discussion

We consider that this work can open the door to new possibilities for face detection and facial analysis. First, we have shown that it is possible to accurately locate facial points with a reduced set of training images and without an expensive annotation of the training data, but simply defining the sought points on the object model. Also, as we use a dense representation of the face, we can easily increase the number of points or experiment with different sets of facial points without any additional cost.

Second, we can use our method for a kind of "inverse cascade" which consist of applying our detector, which is cheap in terms of amount of needed training data and annotation but computationally relatively slow, to an extended set of unannotated training images. In this way we can automatically obtain the annotation of these new images at the level of bounding box and facial points. At this point a faster method which, however, is "expensive" in terms of training data can be trained on the fully annotated data. This is left as future work.

Finally, our method can be used in fine-grained classification (*e.g.* dogs breeds). In this problem, as the similarity between classes is very high, common methods can easily fail. However, it has been shown that using the precise location of some specific parts or points of the object can improve results (*e.g.* dog eyes, paws, tail,..). In this sense, for the moment this is done with a tedious annotation of the location of the object parts. With our method this can be achieved by just learning an object model and then annotate it, exactly as we have shown for faces.

5. Conclusions

In this paper we have presented a new approach for the detection of faces and the localization of their facial points. The approach is based on a deformation field, which is a dense distribution of parts locally interacting with spring-like pairwise connections. During training we learn the deformation model of a face using only the bounding box of a face. Due to a better deformation model, our approach outperforms by a margin all the other methods, even those that use annotated facial points to align the training images. Furthermore, we can visually localize facial points on the trained models and evaluate how the distortions generated by the deformation field can localize these points on test images. Interestingly, this unsupervised technique works quite well and it is even better than some supervised methods.

Acknowledgments

This work was partially supported by Toyota Motor Corporation and FP7 ERC Starting Grant 240530 COGNIMUND.

References

[1] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *in PAMI*, 23(11):1222–1239, 2001. 3, 4

[2] T. F. Cootes, G. J. Edwards, and C. J. Tayolr. Active appearance models. *PAMI*, 23(6), 2001. 2

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, 2005. 2, 5

[4] M. Dantone, J. Gall, G. Fanelli, and L. V. Gool. Real-time facial feature detection using conditional regression forests. In *CVPR*, 2012. 2

[5] M. Everingham, A. Zisserman, C. Williams, and L. Van Gool. The pascal visual obiect classes challenge 2007 (voc2007) results, 2007. 5

[6] M. R. Everingham, J. Sivic, and A. Zisserman. Hello! my name is... buffy" – automatic naming of characters in tv video. In *BMVC*, pages 92.1–92.10, 2006. 2, 6

[7] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, pages 1627–1645, 2010. 1, 2, 3, 4

[8] R. Gross, I. Matthews, J. F. Cohn, T. Kanade, and S. Baker. Multi-pie. *Image and Vision Computing*, 2009. 5

[9] M. hsuan Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE TPAMI*, 24(1), 2002. 2

[10] C. Huang, H. Ai, Y. Li, and S. Lao. High-performance rotation invariant multiview face detection. *IEEE TPAMI*, 29(4):671–686, 2007. 2

[11] G. B. Huang, V. Jain, and E. Learned-Miller. Unsupervised joint alignment of complex images. *ICCV*, 0:1–8, 2007. 2

[12] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments, 2007. 5

[13] M. Lades, J. C. Vorbruggen, J. Buhmann, J. Lange, C. v. d. Malsburg, R. P. Wurtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42(3):300–311, 1993. 2

[14] V. Le, J. Brandt, Z. Lin, L. D. Bourdev, and T. S. Huang. Interactive facial feature localization. In *ECCV*, pages 679–692, 2012. 2

[15] E. G. Learned-Miller. Data driven image models through continuous joint alignment. *IEEE TPAMI*, 28(2):236–250, Feb. 2006. 2

[16] T. K. Leung, M. Burl, and P. Perona. Finding faces in cluttered scenes using random labeled graph matching. In *ICCV*, 1995. 2

[17] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 60(2), 2004. 2

[18] M. Pedersoli, R. Timofte, T. Tuytelaars, and L. V. Gool. An elastic deformation field model for object detection and tracking. Technical Report 1401, KU Leuven/ESAT/PSI, Belgium, 2014. http://homes.esat.kuleuven.be/~mpederso/. 2, 3, 4

[19] J. M. Saragih, S. Lucey, and J. F. Cohn. Deformable model fitting by regularized landmark mean-shift. *IJCV*, 91(2):200–215, 2011. 2, 6

[20] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter. Pegasos: primal estimated sub-gradient solver for svm. *Math. Program.*, 127(1):3–30, 2011. 4

[21] X. Shen, Z. Lin, J. Brandt, and Y. Wu. Detecting and aligning faces by image retrieval. In *CVPR*, pages 3460–3467, 2013. 2

[22] Y. Taigman and L. Wolf. Leveraging billions of faces to overcome performance barriers in unconstrained face recognition. *CoRR*, abs/1108.1122, 2011. 1, 6

[23] Y. Tong, X. Liu, F. W. Wheeler, and P. H. Tu. Semi-supervised facial landmark annotation. *CVIU*, 116(8):922–935, Aug. 2012. 2

[24] M. Ui, V. Franc, and V. Hlav. Detector of facial landmarks learned by the structured output svm. In *VISAPP*, pages 547–556, 2012. 2

[25] M. Valstar, B. Martinez, X. Binefa, and M. Pantic. Facial point detection using boosted regression and graph models. In *CVPR*, pages 2729–2736, June 2010. 2

[26] P. Viola and M. Jones. Robust real-time face detection. *IJCV*, 2004. 2

[27] L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE TPAMI*, 19(7):775–779, 1997. 2

[28] X. Yu, J. Huang, S. Zhang, W. Yan, and D. Metaxas. Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model. In *ICCV*, 2013. 2

[29] A. Yuille, A. Rangarajan, and A. L. Yuille. The concave-convex procedure (cccp). In *Proc. NIPS*, 2002. 4

[30] C. Zhang and Z. Zhang. A survey of recent advances in face detection. Technical report, 2010. 2

[31] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Survey*, pages 399–458, 2003. 2

[32] J. Zhu, L. J. V. Gool, and S. C. H. Hoi. Unsupervised face alignment by robust nonrigid mapping. In *ICCV*, pages 1265–1272, 2009. 2

[33] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, pages 2879–2886. IEEE, 2012. 1, 2, 5, 6