Exploring Shape Models for Facial Landmark Detection

Pierre Sevestre
CentraleSupelec
pierre.sevestre@student.ecp.fr

Louis Ternon
CentraleSupelec
louis.ternon@student.ecp.fr

Abstract

Facial landmark detection is an important intermediary step for many underlying facial analysis operations, from identity recognition to sentiment detection. It aims at detecting and localising characteristic points on people's face. Despite being a simple idea to understand, building robust models has proven challenging due to the great variability of poses and expression, along with other computer vision challenges, such as occlusion or truncation. In this paper, we will dig into a very popular method called Active Shape Model, from [6].

1. Introduction

Establishing facial landmark and facial feature detection are crucial tasks and have a strong impact on underlying computer vision tasks that have to do with faces, such as identity recognition, expression recognition, gesture understanding and 3D animation. We define facial landmark as a set of characteristic points that can serve as anchor point for a 3D animation of one's face. Commonplace landmark are located on the contours of the face, the eyes and the mouse. They are selected in order to be detectable, whatever the facial expression is. Therefore, points such as nose tip and eye corners are very often selected. Main downstream applications of facial landmark are expression analysis, face animation, 3D face reconstruction, head gesture understanding and identity verification. Here is a brief overview of two applications :

- Expression understanding: Facial expressions can be captured through images. They convey emotions and non-spoken messages. For instance, frowning can be interpreted easily as hostility towards someone. The analysis of facial landmark and their temporal evolution can help analyse accurately facial expressions. The study of Facial action unit detection aims at detecting action units from a video of a face - for example, lip tightening and cheek raising. Previous work of their automatic identification was accomplished by Ekman (1978) [12] with his framework of Facial Action Coding Systems, which uses facial landmarks to recognise Action Units, interpret head gestures and recognise facial expressions

- Face recognition : Face recognition techniques usually try to analyse specific regions of interest on the face. When facial landmarks are provided, they locate precisely these areas of interest. They also help measure distances, proportions and angles of the face. See Shi K (2002) [10]

2. Problem definition

Despite many improvements in facial landmark detection, researcher are still working on new algorithms to improve efficiency and robustness. Efficiency is crucial, as new applications need landmark detection algorithms to run in real-time, and to be light enough to be used on embedded devices such as cameras. Robustness is also critical, as applications are used in less marked-out environments and must deal with confounding factors that can hurt the performance of facial landmark detection. Some of them include :

- Variability : From one face to another, factors such as partial occlusions, expression, pose, camera resolution and illumination pose a challenge to algorithm's robustness.

- Acquisition conditions : Similar to recognition, acquisition conditions such as illumination, resolution and background clutter can have a heavy impact on landmark localisation performance. Models trained on a dataset tend to have significantly lower performance when tested on another dataset. Akakin and Sankur (2007) [9] , showed had a performance drop of 20-30% when training their model on FRGC-1 and testing one FRGC-2, two different datasets of the Face Recognition Grand Challenge. FRCG-2 had more diverse conditions than FRGC-1, with varying weather condition, pixel density and occlusions.

Figure 1: Example of possible mouths positions. Smiles, grimaces, facial, hair, rotations and occlusions are recurrent. Good facial landmark detection algorithms should be able to handle robustly these situations as they are frequent in real life.

- Number of landmarks and the expected accuracy. With this expression, we are referring to the fact that there is not one way of deciding which landmarks matter. Datasets can have different numbers of point, with more focus on certain parts of the face than others. For instance, some datasets focus on the eyes, mouth and nose only, whereas other include the contour of the face as well.

Mathematically, the problem can be formulated as follows : Given m training frames, consisting of an image along with its face landmark. A landmark is a set of n points $\{(x_i, y_i)\}$ in d dimension ($d = 2$ in our case). Thus a landmark can be written as $\boldsymbol{x}_j = (x_1, y_1, .., x_n, y_n)^T \in \mathbb{R}^{nd}, j \in [1, m]$.

Then, the problem becomes, given $\boldsymbol{x} = (\boldsymbol{x}_1, .., \boldsymbol{x}_m)$, predict the best landmark location $\boldsymbol{x'}$ on new face image.

3. Related work

3.1. Hand Crafted Models

Flexible models can be elaborated from geometric subcomponents of the face, such as lines, arcs and circles. Yuille et al. [15] modelled parts of the face, mostly eyes and mouth by trying to infer geometric shapes. Even though this model can capture a lot of details, it require a lot of contrast on the image and fails to generalise over different face shapes.

3.2. Articulated Models

Many authors studied articulated models based on rigid components, and connected by sliding or rotating joints. This is approach improves the previous one aforementioned. Beinglass and Wolfson (1991) [2] described a scheme locating geometric parts with Generalised Hough Transform, where the point of articulation is the reference point of each subpart. The connected subparts have to elect the same reference point.

Figure 2: Yuille's approach focuses on geometric shapes. On this image, his models manages to detect the geometric contour of the eye. This is possible due to the high contrast that circles the eye.

3.3. Fourier Series Shape Models

Scott (1987) [14] introduced a method of modelling shapes with trigonometric formulas :

$$x = x_0 + \sum_n a_n \sin n\theta + \psi_n$$

$$y = y_0 + \sum_n b_n \sin n\theta + \phi_n$$

The shape produced is function of parameters a_n, b_n, ψ_n and ϕ_n. Tweaking the parameters enable to produce different types of shaped. Scott's approach tries to search the best parameters to fit the model by minimising an Energy term. Scott's method is innovative, since its formula works on virtually infinite types of shapes and contains no prior shape information.

3.4. Statistical Model of Shape

A substantial literature has studied the distribution of sets of landmark points that describe best an object. Goodall (1991) [7] discussed the use of Procrustes analysis for inferring the mean shape and the covariances between sets of shapes. In a different approach, Grenander et al (1991) developed a method that represents shapes as a set of boundary points connected by arcs. A statistical analysis of the points decided which points were connected by arcs. This model had good results, as it was robust to low quality images. Mardia et al (1991) [11] did something similar: they represented the contours of a shape as a sequence of points, and

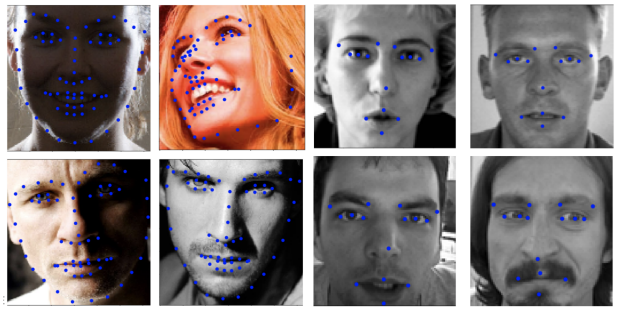

Figure 3: (a) Sample images from 300W dataset [3] , (b) Sample images from Kaggle *Face Images with Marked Landmark Points* [1] challenge

the proximity was described by a covariance matrix. They also try to establish connected neighbouring points via statistical analysis, and Principal Component Analysis (PCA) on the covariance matrix.

4. Methodology

4.1. Dataset

Analysis were conducted using two datasets: *Face Images with Marked Landmark Points* from a Kaggle Competition [1]. There are thousands of images on this dataset. Landmarks are constituted of 15 points. The second dataset is *300 Faces in-the-Wild* (300W) [3] from the iBug. It only has 300 images but they have 68 landmark per frame. Sample images can be found in Figure 3.

4.2. Shape Model

One approach, referred to as *top-bottom* strategy, consists in attempting to fit a prior model of a face to the current sample, as opposed to *bottom-up* strategies, that would consider local patterns, such as edges or corner, to infer a global structure. Hence, one has to create the prior model of the shape, suitable to the tackled problem.

As described in Section 2, one has to build a statistical model using the training landmarks $\boldsymbol{x}$ containing m landmarks consisting of nd points. A Principal Component Analysis (PCA) can be applied to this set of points, projecting any landmark $\boldsymbol{x}_j$ to a space of lower dimension defined by the t eigenvectors with highest eigenvalues retained. The projected vector is denoted as $\boldsymbol{b}_j$.

$$\boldsymbol{b}_j = \Phi^T(\boldsymbol{x}_j - \overline{\boldsymbol{x}})$$

where $\Phi = (\phi_1|\phi_2|..|\phi_t)$, is the concatenation of eigenvectors.

The aforementioned statistical model is the shape model. For facial landmark detection, relevant landmark are position at junction points of the face, such as lips or eyes corners, or areas of high curvature, such as the contour of the face.

Shape alignment Shape should be independent of any particular position, orientation or scale of the face, grasping only intrinsic features, such as inner-face dimensions or facial expressions. It is thus necessary to align shapes into a common co-ordinate frame, prior to any training procedure. A commonly used technique is the Procruste Analysis [8], an iterative optimization procedure aiming at minimizing the Procruste distance, $D = \sqrt{\sum_j (\boldsymbol{x_j} - \overline{\boldsymbol{x}})^2}$.

It can be decomposed into the following steps :

- Arbitrarily choose a reference shape
- Align all shapes to the reference shape, by finding the optimal orthogonal linear transformation
- Compute the mean of the newly created shape set
- If the distance between pre and post alignment shape mean is above a given threshold, re-iterate

Generating plausible shapes The vector **b** defined above contains the parameters of the shape model. Each element b_i is referred to as mode. By varying its value, provided that it stays within the range $\pm 3\sqrt{\lambda_i}$, and projecting the newly obtained vector back to the image instance of the model will generate a new shape for the given mode. It is sometimes possible to comprehend how the given mode behave: example for our two datasets can be found in Figure **??** and Figure **??**.

Modelling local structure Given an image with ground truth landmark $\boldsymbol{X}$, one would like to estimate the likelihood of the set of parameters $(\boldsymbol{b}, X_t, Y_t, s, \theta)$ generating the image instance $\boldsymbol{X}'$. This reside in the choice of a fit function F, such that $F(\boldsymbol{b}, X_t, Y_t, s, \theta)$ would be minimal for the actual model.

An initial solution proposed in [4], corresponding to the case where landmarks are located in strong edges, curvature or intersection, is to use the Euclidean distance between the current point and the strongest edge in the normal direction. That is, if $\boldsymbol{X}$ are the nearest edges, in the normal direction, $F(\boldsymbol{b}, X_t, Y_t, s, \theta) = |\boldsymbol{X} - \boldsymbol{X}'|$.

This methods, although it provided satisfactory results, is not robust, and very sensitive to occlusion in particular.

Another approach consists in using the training set of images to create a descriptor vector distribution of each landmark point. To achieve this, for a given landmark point, one sample n neighboring pixels for each training image

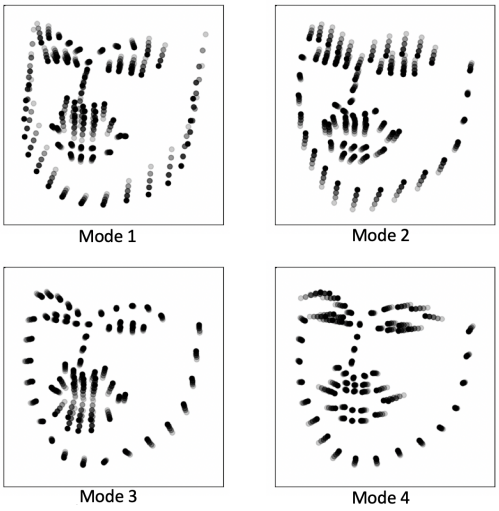

Figure 4: Generating shapes using iBug 300W dataset [**?**]. Varying point intensity account for varying mode value. Modes 1 and 3 correspond to movement of the lips, mode 4 to the extension of the lips and closure of the eyes when smiling, while mode 2 is less obvious.

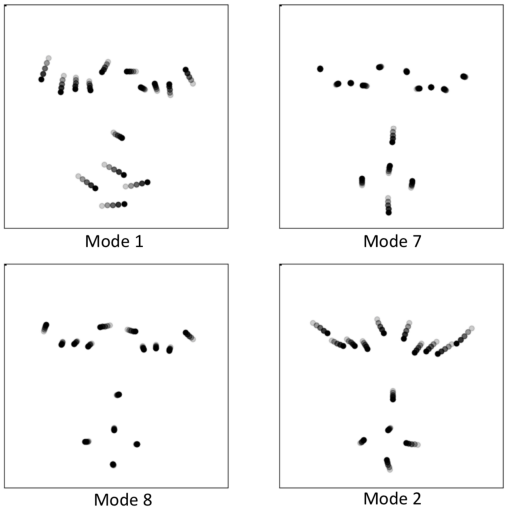

Figure 5: Generating shapes using Kaggle dataset [**?**]. Varying point intensity account for varying mode value. Here again, modes 2 and 8 relate to movement of the eye, 7 of the lips, and first mode is not obvious.

(either 1D profiles, such as in [6] or 2D, such as [13]). Descriptors can be computed on these patches to obtain invariance property to different attributes (from simple gradient, to SIFT, ..). These $m \times n$ patches $\boldsymbol{g}$ are interpreted as multivariate Gaussian for which we compute the average $\overline{\boldsymbol{g}}$ and co-variance matrix Σ_g.

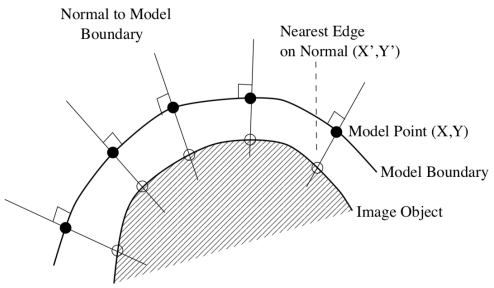

Figure 6: Simple fit function proposed in [6], considering edge with maximum intensity along the normal direction to the model shape

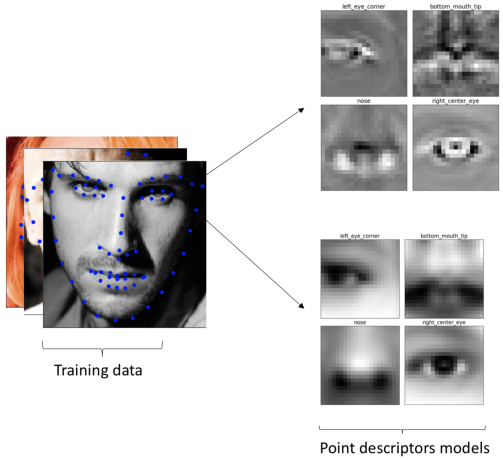

Figure 7: Create statistical model of the landmark neighborhood using training image samples.

Then, a good fit function for a new image instance model $\boldsymbol{X'}, \boldsymbol{g_{X'}}$ for which we derive, is the Mahalanobis distance :

$$D_m(\boldsymbol{g_{X'}}) = (\boldsymbol{g_{X'}} - \overline{\boldsymbol{g}})^T \Sigma_g^{-1} (\boldsymbol{g_{X'}} - \overline{\boldsymbol{g}})$$

4.3. Active shape model

Without prior information about the true localization of the landmarks, minimizing the above fit function is a very complicated optimization task. However, given a rough estimate of the localization (average shape, for example), the Active Shape Model algorithm can be used. It is an iterative algorithm offering good convergence results.

Pseudo-code of the algorithm can be found in 1. In essence, one first explore the neighborhood of each image point $\boldsymbol{X}_i$ to find the best candidate to update your model, $\boldsymbol{X'}_i$ based on the defined fit function. Then you update your model parameters $(X_t, Y_t, \theta, s, \boldsymbol{b})$ to fit the newly found points. This includes multiple underlying processes described later. Finally, we apply constraints of the shape

model to stay in the plausible region, and project back to image level.

Algorithm 1 Active shape model

Result: X, fitting image instance of shape model
Initialize $\boldsymbol{X}$, image instance of shape model
while *norm(* $\boldsymbol{X}$ - $\boldsymbol{X'}$*)* $> \epsilon$ **do**
  for $X_i \in \boldsymbol{X}$ **do**
    $\boldsymbol{X}'_i \leftarrow \min_{x \in N(X_i)} dist(x, descriptor_i)$
  end
  update $(X_t, Y_t, \theta, s, \boldsymbol{b})$ to best fit $\boldsymbol{X}'$
  apply shape constraints on b
  $\boldsymbol{X}' \leftarrow P^T \times \boldsymbol{b}$
end

The ASM algorithm requires to fit a model instance $\boldsymbol{x}$ to a set of image point $\boldsymbol{X'}$. It can be formulated as minimizing $|\boldsymbol{X'} - T_{X_t,Y_t,\theta,s}(\boldsymbol{x})|$, with $\boldsymbol{x}$ lying in the common co-ordinate frame, and T performing the orthogonal transformation (*translation* - *rotation*) and *scaling*. Again, an iterative algorithm is able to provide satisfactory convergence, and is described in [5]. It can be understood as iteratively modelling the shape of the residual between the mean shape $\boldsymbol{x}$ and the target shape **X'** projected in the common co-ordinate frame, $\boldsymbol{x'}$. Finding the best pose parameters require the same alignment strategy as described in Section 4.2.

Algorithm 2 Fitting model to new points

Result: (b, X_t, Y_t, s, θ), best model parameters to match image instance X'
Initialize $\boldsymbol{b}$, shape parameter, to zero.
while *norm(* $\boldsymbol{b}$ - $\boldsymbol{b}'$*)* $> \epsilon$ **do**
  $\boldsymbol{b} \leftarrow \boldsymbol{b}'$
  generate model instance: $\boldsymbol{x} = \overline{\boldsymbol{x}} + \Phi\, \boldsymbol{b}$
  find best pose parameters (X_t, Y_t, θ, s) that best match $\boldsymbol{x}$ to $\boldsymbol{X}'$
  project image points to common co-ordinate frame: $\boldsymbol{x'} = T^{-1}_{X_t,Y_t,s,\theta}(\boldsymbol{X}')$
  project to tangent plane: $\boldsymbol{x}' = \frac{\boldsymbol{x}'}{\boldsymbol{x}'.\overline{\boldsymbol{x}}}$
  retrieve corresponding shape : $\boldsymbol{b}' = \Phi^T(\boldsymbol{x}' - \overline{\boldsymbol{x}})$
  apply shape constraints on $\boldsymbol{b}'$
end

An example of iteration of the algorithm can be found on Figure 8

5. Experiments

We ran our experiments on particular subsets of the datasets presented in Section 4.1. Especially, we only kepts images that contained the all the landmark points, i.e. 15 for the *Kaggle dataset*, 68 for *300W*, resulting in 2140 and 300 frames respectively. These sets were splitting into training - validation and testing set, accounting for 60%, 20% and 20% of the total number of frames.

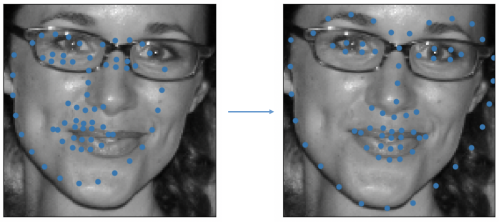

Figure 8: Left: Before ASM iteration, the initial landmark correspond to the mean shape, projected to the image frame. Right: landmark after 4 iterations

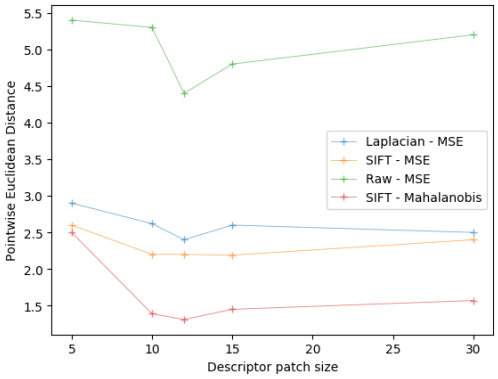

Figure 9: Experiment results on *Kaggle dataset* [**?**]. PED with respect to descriptor patch size, for multiple strategies

The evaluation metric used is the Point-wise Euclidean Distance (PED), formulated as : $PED(x, y) = \frac{1}{n} \sum_{i=1}^{n} d(x_i, y_i)$, d being the euclidean distance.

Kaggle Dataset Multiple experiments were performed on the *Kaggle dataset* to study the effect of the different parameters of the model. In particular, evaluation was performed on :

- Descriptor type (*Laplacian*, *SIFT*, None)
- Fit function (*Mean squarred Error* - MSE, *Mahalanobis* distance)
- Descriptor patch size (Ranging from 5 to 30)
- Neighborhood search range (Ranging from 5 to 30)

Results can be found in Figure 9. As expected, best results are obtained using *SIFT* descriptors along with *Mahalanobis* distance. However, for descriptor created using small patches, both fit function are almost equivalent.

Laplacian filtering performs comperatively to *SIFT* when MSE is used as fit function. It was not possible to apply Mahalanobis distance with Laplacian, the variance of the descriptor was too small, leading to exploding value in distance computation.

An optimal value for the descriptor patch size seems to be around 10 to 15.

Sample of the predictions can be found in Figure 12. Results are very good, with 1.2 pixels error on average, even in complicated cases, with occlusion from glasses, moustache, or extreme poses, such as closed eyes.

300W Dataset Results summary of our experiments can be found in Table 1. The final PED is significantly greater on this dataset, and the performance seems to be consistent over the different combination of parameters. However, it is hiding very different underlying behavior.

As it can be seen in Figure 11, using a great descriptor of important size (e.g. 30, i.e 61×61 grid) tend to have high variance in predictions. Some landmark achieve remarkable accuracy, while other fails to converge. On the opposite, smaller patches will have more consistent results. Additional results can be found on Figure **??**.

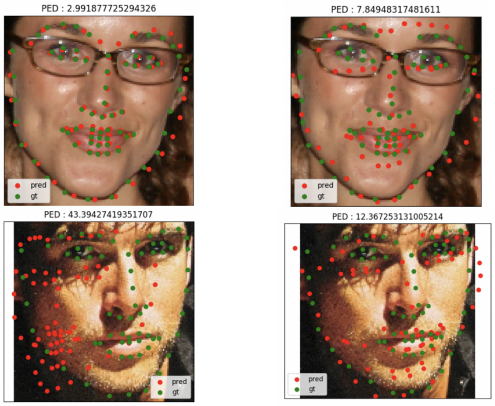

Figure 11: Left: Prediction made using patch of size 30. Right : Prediction made using patch of size 12. - While descriptor computed using important patch size perform remarkably on some image, it also tend not to converge on others, whereas prediction with smaller descriptors have more consistent results.

Conclusion

In simple situations, where the face is already oriented toward the camera, without important occlusion and pronounced landmark localization, shape models perform very well, even with simplistic descriptors. However, when the task increases in complexity, with greater variability, heavy occlusion, unusual poses, then it is often not sufficient to get satisfactory results. To go further and improve our model to tackle these shortcomings, we could enhance the descriptor used, considering multi-scale, multi-view solutions, or couple the shape model with appearance model.

References

[1] 3

[2] W. H. Beinglass A. Articulated object recognition, or, how to generalize the generalized hough transform. *oc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1991. 2

[3] S. Z. M. P. C. Sagonas, G. Tzimiropoulos. 300 faces in the wild challenge: the first landmark localization challenge. *Proceedings of IEEE, International Conference on Computer Vision (ICCV-W), Workshop on 300 Faces in the Wild Challenge (300-W)*, pages 397–403, 2013. 3, 7

[4] T. Cootes. An introduction to active shape models. *Image Processing and Analysis, Model-Based Methods in Analysis of Biomedical Images*, pages 223–248, 2000. 3

[5] T. Cootes and C.J.Taylor. Statistical models of appearance for computer vision. *Imaging Science and Biomedical Engineering*. 5

[6] T. Cootes and C.J.Taylor. Active shape models - their training and their application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995. 1, 4

[7] C. Goodall. Procrustes methods in the statistical analysis of shape. *3rd Alvey Visio Conference, Cambridge*, 1991. 2

[8] C. Goodall. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society*, (53):285–339, 1991. 3

[9] A. HC and S. B. Robust 2d/3d face landmarking. *Proc. of Conf. on 3DTV. Kos*, 2007. 1

[10] S. J, S. A, and M. D. How effective are landmarks and their geometry for face recognition? *Comput. Vis. Image Understand*, 2006. 1

[11] J. K. K.V. Mardia and A. Walder. Statistical shape models in image analysis. *oc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1989. 2

[12] E. P and F. WV. Facial action coding system: A technique for the measurement of facial movement. *Palo Alto: Consulting Psychologists Press*, 2003. 1

[13] F. N. S. Milborrow. Active shape model with sift descriptors and mars. *In Press*, 2013. 4

[14] G. L. Scott. The alternative snake - and other main animals. *Alvey 3rd Visio Conference, Cambridge*, 1989. 2

[15] H. P. Yuille AL, Cohen D. Feature extraction from faces using deformable templates. *In Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition. San Diego, CA, USA*, 1989. 2

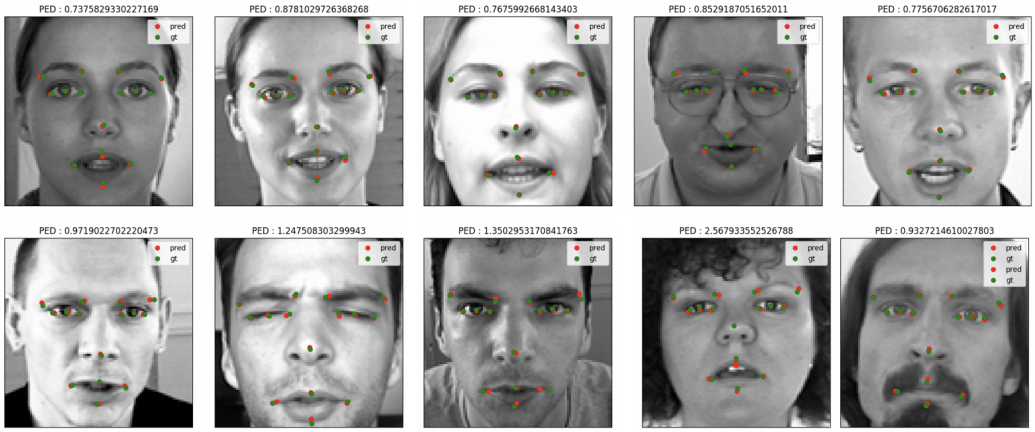

Figure 10: Sampling of the results obtained on *Kaggle dataset* [**?**], using SIFT descriptors, with Mahalanobis distance, and a descriptor patch size of 12.

Descriptor Patch Size	Neighbor search range	Fit function	Descriptor	PED	Execution Time (s per frame)
5	35	Mahalanobis	SIFT	16.3	89
10	25	Mahalanobis	SIFT	15.1	58
12	25	Mahalanobis	SIFT	14.4	91
12	25	Mahalanobis	Laplacian	24	**20**
20	30	Mahalanobis	SIFT	**14.2**	132
30	20	Mahalanobis	SIFT	14.8	116

Table 1: Results obtained on *300W dataset* [3] for various parameters configuration. PED averaged over 20 images.

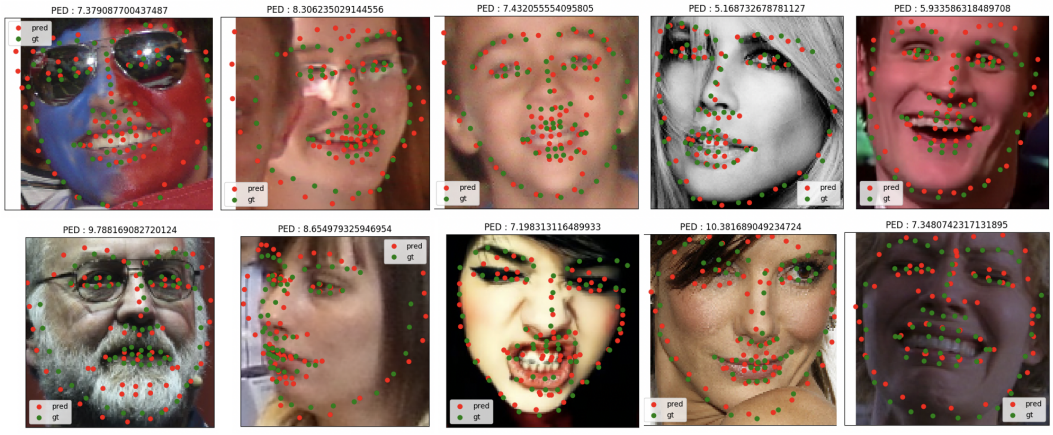

Figure 12: Sampling of the results obtained on *300W dataset* [3], using SIFT descriptors, with Mahalanobis distance, and a descriptor patch size of 30.