Implementing Cascade of Regression-based Face Landmarking: an in-Depth Overview

Romuald Perrot, Pascal Bourdon, David Helbert

Implementing cascaded regression tree-based face landmarking: an in-depth overview

Romuald Perrot, Pascal Bourdon, David Helbert*

Abstract

Face landmarking, defined as the detection of fiducial points on faces, has received a lot of attention over the last two decades within the computer vision community. While research literature documents major advances using state-of-art deep convolutional neural networks, earlier *cascaded regression tree*-based approaches remain a relevant alternative for low-cost, low-power embedded systems. Yet, from a practical point of view, their implementation and parametrization can be a difficult and tedious process. In this paper, we provide the readers with insights and advice on how to design a successful face landmarking system using a cascade of regression trees.

Keywords: Face landmarking, Cascaded Regression, Regression trees, Temporal tracking

1 Introduction

Automatic landmarking is a mandatory step for many high-level face analysis applications such as person identification, expression transfer or emotion recognition. As a result, it has attracted a lot of research efforts over the last two decades, with regular claims of either high temporal performance (speed), or high accuracy (task precision). High temporal performance on limited hardware is reportedly achieved using the state-of-art *Cascaded Regression* (CR) framework. However in many publications a lot of technical details are omitted, making its implementation a difficult and tedious task. In this paper, we aim at analysing the practical aspects of one type of CR-based face landmarking, namely the Cascaded Regression Trees (CTR), and how to implement it successfully for real-life video sequences. We investigate both the algorithmic angle and the effects of parametrization on results, based on the most relevant contributions one can find in research literature.

The article is organized as follows: in section 2 we review the most well-known research work on face alignment and landmarking. In section 3, we provide the theoretical background that lies behind cascade of regression trees algorithms. In section 4, we detail each part of the actual implementation of the landmarking process. In section 5, we introduce enhancements for face landmarking in videos. In section 6, we provide experimental results to explore the effects of parametrization landmarking quality, and set a performance benchmark of our implementation using the well-known 300W competition test-set [1, 2]. This section is followed by a discussion in section 7, before section 8 finally draws the conclusion of this work.

2 Related work

The objective of this section is first to present a brief overview of the current state-of-the-art in face landmarking and, second, to introduce the reference articles about CR-based methods that are needed to understand this work. Detailed overviews about either the first or second topics can be found in surveys such as [3, 4] or [5, 6], respectively.

Many face landmarking techniques have been proposed over the years. It is a common practice to split these into two main categories: generative and discriminative models. Generative approaches

*Univ. Poitiers, CNRS, XLIM, UMR 7252, F-86000 Poitiers, France
I3M, Common Laboratory CNRS-Siemens, University and Hospital of Poitiers, France
david.helbert@univ-poitiers.fr

rely on a analysis-by-synthesis philosophy, where a system models the generative processes by which face images are formed using a given number of parameters. Analysis is then the search for a set of parameter values that best explained an observed image in terms of this synthesis, in our case landmark positions. Notable examples of this category are Active Appearance Models (AAM) [7], where global face texture and shape variations are captured by Principal Component Analysis (PCA), or Gauss-Newton Deformable Part Models (GN-DPM) [8], which are based on local texture models instead. Discriminative approaches on the other hand rely on directly looking for relevant (discriminative) information regarding what should be the position of a landmark (output) given the observed image (input). Discriminative face landmarking techniques include Constrained Local Models (CLM) [9, 10], Cascaded Regression (CR) [11, 12, 13, 14, 15], Supervised Descent Methods (SDM) [16, 17, 18, 19], and of course Convolutional Neural Networks (CNN) [20, 21, 22]. All said methods share some level of similarities. As an example, both CLM and GN-DPM define local texture analysis models around each landmark, while AAM and CNN analyse face textures as a whole, with frequent criticism on global texture models driving the fitting process as a whole and either lacking robustness to occlusions and light changes or leading to highly complex models. Several methods rely on transitional feature spaces such as Local Binary Patterns (LBP), Scale Invariant Feature Transforms (SIFT), pixel differences, or convolutional kernels. And finally many rely on so-called *cascaded regression* models, which are basically data-driven versions of classical gradient descent-based iterative alignment techniques based on the famous Lucas-Kanade tracker [23, 24]. A theoretical analysis of data-driven cascaded linear regression from the perspective of least squares optimization is provided by supervised descent methodology in [17].

As of today, Cascaded Regression is still one of the most popular methods employed for face landmarking when low computational resources are required. While deep architectures with mixed linear/non-linear stages (*i.e.* CNN) have been found highly effective for this task in terms of accuracy, their strong reliance on parallel computation efficiency and intensive use of Graphics Processing Unit (GPU) resources prevent them from being used on low-cost, low-power embedded systems. It is important to stress out that our most of our research work on face analysis is indeed aimed at such systems. Regression-based methods are claimed to enable both high-performance and high-robustness in face landmarking, even on limited hardware. Dollar *et al.* [11] introduced the cascaded pose regression method, where a shape is progressively refined to a target shape. Cao *et al.* [12] enhance regression using a new shape indexation scheme and a boosted two-level cascade of regression. Kazemi *et al.* [13] provide a faster regression system through a simplified initialization scheme and a gradient boosted cascade building. Ren *et al.* [14] announce a three times speed-up using cascaded linear regression on a tree-based feature space called *local binary features* and achieve face alignment at 3000fps on a single-core desktop and 300fps on a modern mobile phone. This method can be seen as merging the best of two worlds, namely linear and non-linear (*i.e.* tree-based) regression. Purely linear methods usually fail to learn the complex relationship between feature spaces and shape variations, resulting in poor fitting capabilities in unconstrained scenarios on a global basis. Non-linear methods however tend to show better fitting capabilities, often with straight-forward, time-effective computations, at the risk of exhibiting sudden and violent failure in very specific cases. Because our primary focus is to achieve face landmarking using the lowest computational resources, we decided to focus on tree-based CR methods using pixel difference feature spaces.

3 Theoretical Background

3.1 Random tree

Regression tree-based face landmarking heavily relies on supervised machine learning with *random tree* models [25]. We denote by *splitting criterion* the internal nodes which contain decisions. This criterion splits samples into two disjoint subsets, and leaves containing a displacement vector. The tree depth is generally defined constant for all trees.

For a given node, a splitting function indicates the belonging of a sample to a subtree. For any given sample $\mathbf{x}_i$, the splitting decision result is defined as $k = \phi(\mathbf{x}_i)$, with $k \in [l, r]$, l and r are respectively for the left and the right subtree. Function $\phi(.)$ behaves as a similarity measure between

two descriptions (*i.e. features*) $\mathcal{F}_1$ and $\mathcal{F}_2$ using threshold κ:

$$\phi(\mathbf{x}) = \begin{cases} l & \text{if} \quad d(\mathcal{F}_1, \mathcal{F}_2) > \kappa \\ r & \text{else.} \end{cases} \tag{1}$$

with $d(.)$ is a distance operator as subtraction or Euclidean norm. In our model, each node has its own splitting function.

3.2 Face alignment

Face landmarks define a global semantic model we refer to as the *shape*. In our study we will focus on 2D shapes. The standard way to perform shape alignment over a face image is to use a two-step approach: detection then regression. Face detection can be performed using the popular Viola and Jones detector [26]. The output of this first process is a bounding box which provides a rough approximation of face position and scale and serves as an initialization step for the regression process. Regression then progressively refines shape location and deformation during a second step to match the target face.

3.3 Training set

A training set $\mathcal{T}$ is required to be a reference for landmarking performance and to build the model. Such a set is composed of images and their ground truth annotations (*i.e.* shape). Its building is generally a tedious manual process which consists in indicating on each image the locus of every landmark which composes the shape. We can use publicly available datasets such as [2, 27, 28] if landmark indices and order are consistent across images.

Let $\mathcal{I}_i$ be the image and $\mathbf{y}_i$ be the ground truth annotation. We denote by $\mathbf{x}_i = (\mathcal{I}_i, \mathbf{y}_i)$ the i-th sample of a training set, and build an *augmented set* $\mathcal{T}' = \{\mathbf{x}'_i\}$ using synthesized perturbations $\mathbf{y}'_i$ on the top of a training set $\mathcal{T} = \{\mathbf{x}_i\}$. The residual $\mathbf{r}_i$ of a sample $\mathbf{x}_i$ is defined as the difference between ground truth and noisy annotations:

$$\mathbf{r}_i = \mathbf{y}_i - \mathbf{y}'_i. \tag{2}$$

4 Cascade of regression-based face alignment

4.1 Cascade of regression

The main idea of face landmarking using regression is to iteratively refine a shape $\mathcal{S}^t$ to an optimal target shape according to an input image $\mathcal{I}$, a single regression model $\mathcal{R}_t$, and a previous state $\mathcal{S}^{t-1}$. We can write this process as:

$$\mathcal{S}^t \leftarrow \mathcal{S}^{t-1} + \mathcal{R}_t(\mathcal{I}, \mathcal{S}^{t-1}). \tag{3}$$

The complete regression model $\mathcal{R}$ is the accumulation of all single regression models $\mathcal{R}_t$, which can differ from one another. In the case of our study, a single regression model is a tree, and the increment for refinement will be computed from an input image and a shape. Cascades of regressions are relied by current regression approaches. It is a two-level variation of the regression method: K trees compose a single cascade and a common feature set is shared by all trees within this same cascade (see Figure 1). If the cascade number is T, the total iteration number in the regression process equals $T \times K$. in In [13] Kazemi suggests that regression should start using the mean shape of the training set as an initial shape $\mathcal{S}^0$.

4.2 Gradient boosting

The underlying method used to learn the face regression model is defined as *gradient boosting*. It combines boosting with gradient descent yielding as a robust and efficient training process that converges

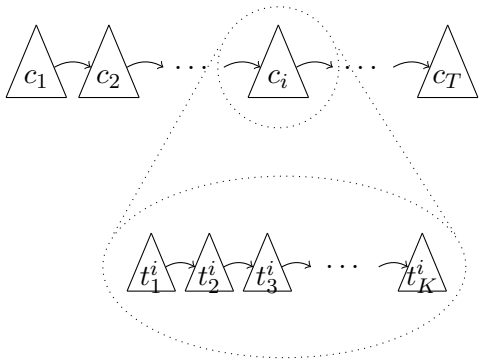


Figure 1: Regression cascade: cascade c_i is a sequence of K regression trees t^i_j.

to the correct solution while enabling generalization of the model to unknown data. Here we sum up the formulation, given by Breimann [25].

Let C be the cost function which estimates the error rate of a regressor $\mathcal{R}$ over a sample $\mathbf{x}_i$:

$$C\left(\mathcal{R}(\mathbf{x}_i), \mathbf{y}_i\right). \tag{4}$$

The training loss $\mathcal{C}$ over all samples of the training set is defined as:

$$\mathcal{C}(\mathcal{R}) = \sum_i C(\mathcal{R}(\mathbf{x}_i), \mathbf{y}_i). \tag{5}$$

The goal of the training phase is to find a model that minimizes this loss:

$$\mathcal{R}_{opt} = \arg\min_{\mathcal{R}} \mathcal{C}(\mathcal{R}). \tag{6}$$

Assuming we have an additive regression model with $T-1$ submodels:

$$\mathcal{R}^{T-1} = \sum_{t=1}^{T-1} \mathcal{R}_t. \tag{7}$$

The problem is how to find the next increment to $\mathcal{R}^T$ with respect to $\mathcal{R}_{opt}$:

$$\mathcal{R}^T = \mathcal{R}^{T-1} + \Delta \tag{8}$$

In a gradient boosting approach, update parameter Δ is the result of a new regression model fitted on the gradient. During step T, we will compute a regression model $\mathcal{R}_T$ that minimizes gradient values:

$$\mathcal{R}_T = \arg\min_{\mathcal{R}} \mathcal{C}(\mathcal{R}(\mathbf{g}_T), 0). \tag{9}$$

where $\mathbf{g}_T$ is the gradient value of the training loss at beginning of stage T:

$$\mathbf{g}_T = \frac{\partial C(\mathcal{R}_{T-1}(\mathbf{x}_i), \mathbf{y}_i)}{\partial \mathcal{R}_{T-1}(\mathbf{x}_i)} \tag{10}$$

Intuitively, gradient value computed so far serve as input to drive the construction of the new tree. Finally, the update value Δ is the result of that tree:

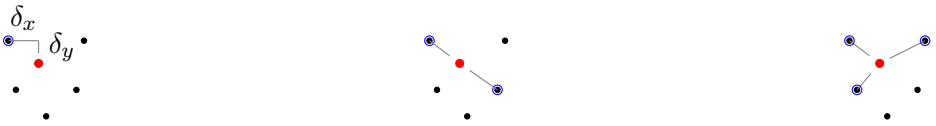
$$\Delta = \mathcal{R}_T \tag{11}$$

4.3 Feature indexation

When it comes to face landmarking, the features used in a tree node splitting functions usually define point positions within a reference shape $\mathcal{S}_{ref}$, resulting in multiple pixel intensity comparisons (*pixel differences*) being used as descriptors. Various indexation schemes have been investigated to ensure robustness during the fitting process. Point positions can be (see Figure 2):

- Shape-indexed (SI) [12], when features are translation vectors relative to the closest landmark;
- Interpolation shape-indexed (ISI) [29] when features are linear interpolations between two landmarks;
- Barycentric interpolation shape-indexed (BSI) [30], when features are barycentric interpolations of three landmarks.

Shape indexation is the simplest form of description, unfortunately it is not robust against rotation. Interpolation shape indexation is, but the features must be located on a line between two landmarks, thus limiting the feature subspace heavily. Barycentric interpolation is the most flexible solution and will be used during our experiments. Yang *et al* [31] proposed another barycentric indexation scheme based on three random landmarks instead of the three nearest ones in Cao *et al*'s [30] work.



(a) SI (b) ISI (c) BSI

Figure 2: Feature indexation

4.4 Splitting function

As mentioned in the previous section, splitting functions $\phi(.)$ usually consist in simple computations of pixel differences in the grayscale space. These can be written as :

$$\phi(\mathbf{x}) = \begin{cases} l & \text{if} \quad \mathcal{I}(\mathcal{F}_1) - \mathcal{I}(\mathcal{F}_2) > \kappa \\ r & \text{else.} \end{cases} \tag{12}$$

where features $\mathcal{F}_1$ and $\mathcal{F}_2$ are converted from point positions within the reference shape $\mathcal{S}_{ref}$ to pixel positions within the target image $\mathcal{I}$.

In a classical implementation, splitting decision at each node consists in thresholding the difference of intensity values at a pair of pixels, which allows relative insensitivity to changes in global lighting compared to single-pixel thresholding. Pixel differences can be computed directly on raw image data without any pre-processing whatsoever such as gaussian kernel-induced regularization or feature-space transformation (LBP, BRIEF, SIFT, ...), resulting in faster computation. The use of more robust, fitted-for-the-task feature spaces has been and should remain investigated, as their degree of specialization can help reducing tree depths and the number of cascade iterations. In our case, we decided to use simple differences on gray levels and expect barycentric interpolation to induce robustness and specialization.

4.5 Training process

In this section we sum up the training process of the cascade of regression. First we detail how a regression model $\mathcal{R}_T$ is estimated over an augmented training set $\mathcal{T}'$, then how cascades of regression are trained.

We describe the gradient boosting approach proposed by Kazemi *et al.* [13], which is one of the most efficient and easy to implement. A recursive top-down approach using estimation residuals is used to build each tree in the cascade to drive construction phases.

Node training Let Φ be a set of splitting criteria, the aim is to find the best ϕ_{opt} that divides the training set into two subsets $\mathcal{T}_l$ and $\mathcal{T}_r$ such that residuals of samples belonging to a same subtree leads to a unique value. Intuitively, samples sharing similar face characteristics should belong to a same subtree. The following energy function has thus to be minimized:

$$E(\phi, \mathcal{T}') = \sum_{k \in \{l,r\}} \sum_{\phi \in \Phi} \|\mathbf{r}_i - \mu_{\phi,k}\|^2 \tag{13}$$

with $\mu_{\phi,k}$ the mean value of residuals for every sample i belonging to a same subtree k:

$$\mu_{\phi,k} = \frac{1}{|\mathcal{K}|} \sum_{i \in \mathcal{K}} \mathbf{r}_i \quad \text{with} \quad \mathcal{K} = \{i / \phi(\mathbf{x}_i) = k\}. \tag{14}$$

The implementation consists in choosing S splitting criteria $\phi_j \in \Phi$, and selecting the one that minimizes energy E:

$$\phi_{opt} = \arg\min_{\phi_j} E(\phi_j, \mathcal{T}'). \tag{15}$$

Kazemi *et al.* [13] suggest the computation of a further derivation to reduce computation costs by avoiding a complete re-evaluation of some terms in energy function E. A selected criterion splits the training set into two disjoint sets, before the optimization process with its corresponding subset is applied to train each subtree of a newly built node.

Leaf value A leaf is created when a maximum tree depth criterion is reached. Its value is calculated from the mean of residuals for samples reaching this leaf. A shrinking approach is usually employed by authors in order to avoid overfitting issues and reduce the influence of noisy data. The value of leaf $\mathcal{L}_k$ is thus obtained by:

$$\mathcal{L}_k = \frac{\alpha}{|\mathcal{K}|} \sum_{i \in \mathcal{K}} \mathbf{r}_i \tag{16}$$

where α is a shrinking factor which can be either defined as a constant [13] or computed for normalization and stability purposes [12].

Cascade training Training of the regression cascade is simply a sequence of tree training. The single modification is the adjustment of residuals before computing a new tree. Every training sample $\mathbf{x}'_i$ is updated such that:

$$\mathbf{y}'_i \leftarrow \mathbf{y}'_i + \mathcal{R}_t(\mathcal{I}_i, \mathbf{y}'_i). \tag{17}$$

5 From face alignment to face tracking

A common issue with standard two-step face regression is its strong dependency on face detection settings: (i) a small perturbation on the initial face bounding box can cause large changes on the outcome of the regressor (ii) this dependency sometimes makes it mandatory that the detection algorithm used during training be the same as the one used during online regression [32]. In video tracking this two-step approach becomes a computational waste because previous detection or tracking results from a given frame n cannot be used as a plausible initial guess for the next frame $n+1$. In this section we detail some solutions to provide an effective face tracking solution.

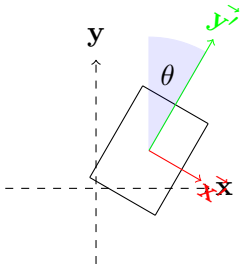


Figure 3: Oriented bounding box definition. The angle of rotation between the local frame $(\vec{x'}, \vec{y'})$ and the global frame $(\vec{x}, \vec{y})$ is θ.

5.1 Data augmentation

To emulate the risk of misplaced initial bounding boxes we first work on the data augmentation process. Let's recall that for a sample $\mathbf{x}_i = \{\mathcal{I}_i, \mathbf{y}_i\}$ of the training set $\mathcal{T}$, this step produces an augmented sample $\mathbf{x}'_i = \{\mathcal{I}_i, \mathbf{y}_i, \mathbf{y}'_i\}$. Reference face regression method [12] proposed to randomly select the ground truth shape of a sample j as the noisy (augmented) shape for another sample i. We decide to use the same augmentation scheme while adding rigid geometric perturbations. In practice, we first generate an augmented shape $\mathbf{y}'_i$, then apply a random rigid transformation:

$$\mathbf{y}'^{*}_i \leftarrow s_\xi \cdot R_\xi \mathbf{y}'_i + \Delta_\xi \tag{18}$$

where s_ξ is a scale factor, R_ξ a 2D in-plane rotation centered on the initial bounding box and Δ_ξ a displacement. This is a generalization of Ren *et al.*'s proposal [14], which only introduces translation. Each parameter is uniformly sampled according to their respective range, which are $[60\%; 140\%]$ for scaling, $[-\frac{\pi}{3}; \frac{\pi}{3}]$ for rotation, and $[-50\%; 50\%]$ for translation relative to the size of the bounding box (translation in x and in y are independent, for instance, δ_x is $\pm 25\%$ of the width). Such ranges were chosen as a reasonable trade-off between fitting precision and robustness.

5.2 Frame to frame tracking

Given the data augmentation scheme, on video segments with smooth motion, estimated shapes should provide accurate initial bounding boxes for the next estimation on a frame-per-frame basis, which is a common strategy in generative face alignment. Unfortunately, should face fitting fail, the resulting bounding box may compromise the next estimation to come. Moreover, face detection only returns an axis-aligned bounding box, which becomes an issue in mobile devices where large in-plane rotations can be observed. We propose two approaches to enhance robustness in such cases.

Median box filtering In most situations, face motion tends to be smooth enough to introduce inter-frame redundancy we can use for filtering purposes. Based on this hypothesis, we compute a set of independent median values for shape mean position, width and height from a given prior estimations to serve as the reference for initial bounding box parameters.

Oriented frame We remove in-plane rotation caused by the camera system using an oriented bounding box where the main axis is as close as possible to the symmetric axis of the face being tracked. This bounding box defines a standard local frame in which a face is always forehead at top and chin at the bottom. Our oriented bounding box is characterized by an axis aligned bounding box and a rotation angle relative to the center of the bounding box (see Figure 3).

In practice, following face detection, we set the orientation angle to 0, proceed with the alignment process, then compute the relative orientation using a Procrustes analysis between the fitted shape and the mean shape. Note that the local frame is integrated into the features of the regression model. This leads to a lower computational cost than applying rotation on the input image and it doesn't suffer from discretization effect.

5.3 Regularization

In regression based face alignment, the result of the fitting process can be temporally noisy. This is particularly noticeable on video sequences where a target face stays still yet the estimated landmarks appear to be in fast, noisy motion. This effect is mainly due to two factors: noise in the image and the simplicity of the splitting node decision (*i.e.* simple pixel difference). We propose two complementary steps to reduce noise in temporal tracking.

Shape filtering Similar to the bounding box smoothing, we do not use the last shape as the result of the fitting process but a combination of the p preceding shapes. We use a modified and simplified energy minimization algorithm as proposed by Cao [30] in a context of 3D face regression. Given the last two fitted shapes, $\mathcal{S}^{n-1}$, $\mathcal{S}^{n-2}$, the current output of the regressor $\hat{\mathcal{S}}^n$, we compute the regularized shape $\mathcal{S}^n$ by minimizing the following energy term:

$$E = w_{reg} \cdot E_{reg} + w_{tm} \cdot E_{tm} \tag{19}$$

where:

$$E_{reg} = \|\mathcal{S}^n - \hat{\mathcal{S}}^n\|^2 \tag{20}$$

$$E_{tm} = \|\mathcal{S}^{n-2} - 2\mathcal{S}^{n-1} + \mathcal{S}^n\|^2 \tag{21}$$

E_{reg} is a regularization term that restrains the corrected shape from drifting too far away from the initial estimation given by the regressor, while E_{tm} is a temporal constraint which ensures smoothness over the last three fitted shapes.

Weighted interpolation Shape filtering still does not solve all stability issues. Additionally, we bring a further step using a weighted linear interpolation of the last p preceding shapes:

$$\mathcal{S}^n_{final} = \sum_{i=1}^{p} \alpha_i \mathcal{S}^{n-i} \tag{22}$$

where α_i is the weight associated with the shape at frame $n-i$ (with $\sum_{i=1}^{p} \alpha_i = 1$). We define the values the weights using the variance between the image at frame n and image at frame $n-1$. Intuitively, if the variance is small, the shape should be similar to the preceding and should not receive strong importance.

Additionaly if the variance is higher than a predefined threshold, we clear the stack used for the interpolation since it indicates presence of a new key image (*i.e.* a large change in face expression or face position). Since variance is related to the capture system, the threshold value is also capture system specific.

6 Experimental results

In this section, we present experimental results and investigate the influence of parametrization. First we detail the methodology then we review the influence of each relevant parameter on result quality. Additionally, we also compare the performance of our implementation on the challenging test-set of 300W competition [1, 2] in terms of cumulative error curves, Area-Under-Curve (AUC) values, failure rate and computational time.

6.1 Training data and methodology

The training set is generated using the two following publicly available face datasets provided by Cao *et al.* (`http://gaps-zju.org/DDE/+`):

- LFW [33] provided by Huang *et al.* `http://vis-www.cs.umass.edu/lfw/`,

- FaceWarehouse [34] provided by Cao *et al.* `http://kunzhou.net/zjugaps/facewarehouse/` (3D annotations have be reduced to 2D projections).

FaceWarehouse is made of controlled and strict face acquisitions, while LFW presents labeled faces in the wild. The training set is composed of 6400 random samples taken from the two sets, with 1600 remaining samples being used as the testing set. There is no overlap between both sets.

Landmark position errors are defined as the mean distances between fitted shapes and their ground truth counterparts. To limit the impact of spatial resolution differences between testing samples, all results are normalized with respect to the intra-ocular distance, as literature usually suggests. During training, the bounding box used for initialization is not the result of a face detection algorithm. Indeed it is the minimal one that encompasses ground truth shapes.

6.2 Terminology and notations

Regression models depends are parametrized by:

- T the number of cascades ;
- K the number of trees in a cascade ;
- S the number of augmented samples per training sample ;
- R the number of splitting functions candidates ;
- P the number of features generated per cascade.

Two additional parameters are used to introduce perturbations onto the training set during data augmentation. In all tests, these are set to: $\Delta_s = \pm 40\%$, $\Delta_x, \Delta_y = \pm 50\%$.

6.3 Influence of parameters

We study the impact of all parameters using the median square error of all regression results. For each parameter to be investigated, all others are set to their default value which are: $T = 10$, $K = 500$, $S = 300$, $P = 500$, $R = 50$.

Influence of T We start by studying the impact of T, the number of cascades in the regression. Figure 4 shows median square error evolution for $T \in [1; 20]$.

At first we can notice a large amount of error reduction from $T = 1$ to $T = 10$, then from $T = 10$ this reduction starts to slow down, thus justifying the use of $T = 10$ as default values.

Influence of K Parameter K represents the number of trees per cascade. Figure 5 shows its impact on result quality for $K \in [50; 1000]$.

Error reduction in the case of higher K values is not as significant as it was with T values. While a value of $K = 200$ seems to be a good precision/computational cost trade-off, we later noticed that this result is most likely a side-effect of training set over-fitting, as poor generalization was usually observed in real-life landmarking situations (*e.g.* webcam tracking). Authors usually suggest $K = 500$ as an optimal value, but there is little evidence this choice is the best, as it heavily depends on size of the training set and performance expectations. Still, $K = 500$ appears to be a reasonable choice and to align ourselves with state-of-art methods, we will use this value for the next experiments.

Influence of P Parameter P sets the number of features generated per cascade. Figure 6 shows median errors for $P \in [50; 1000]$.

Once again, landmarking errors decrease slowly with higher P values, with a good trade-off being $P = 500$. Note that compared to T and K, this parameter only impacts training time and has no effect whatsoever on fitting time performance.

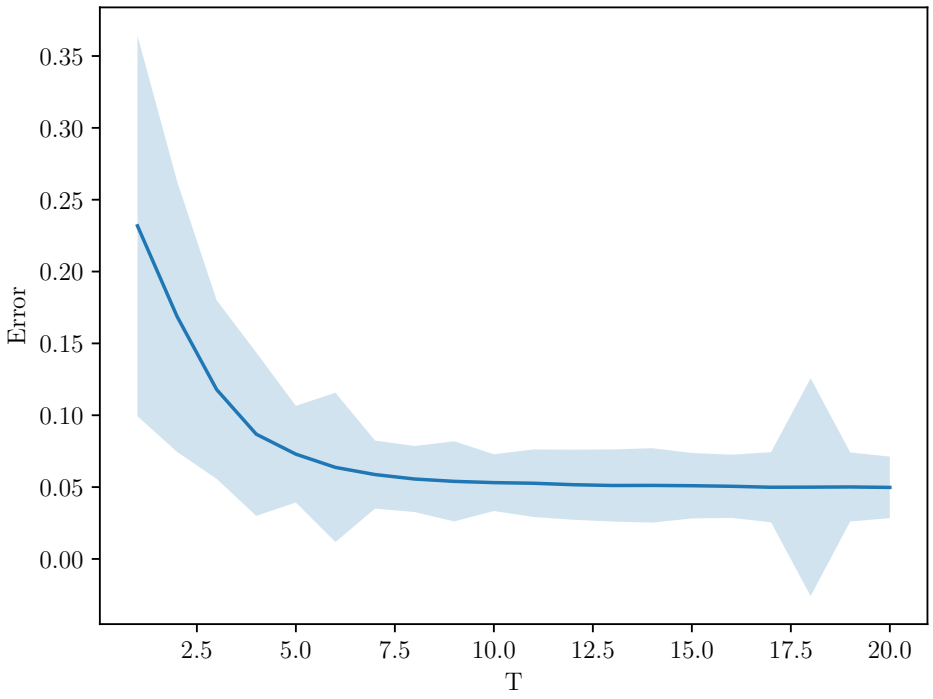


Figure 4: Parameter $T \in [1; 20]$

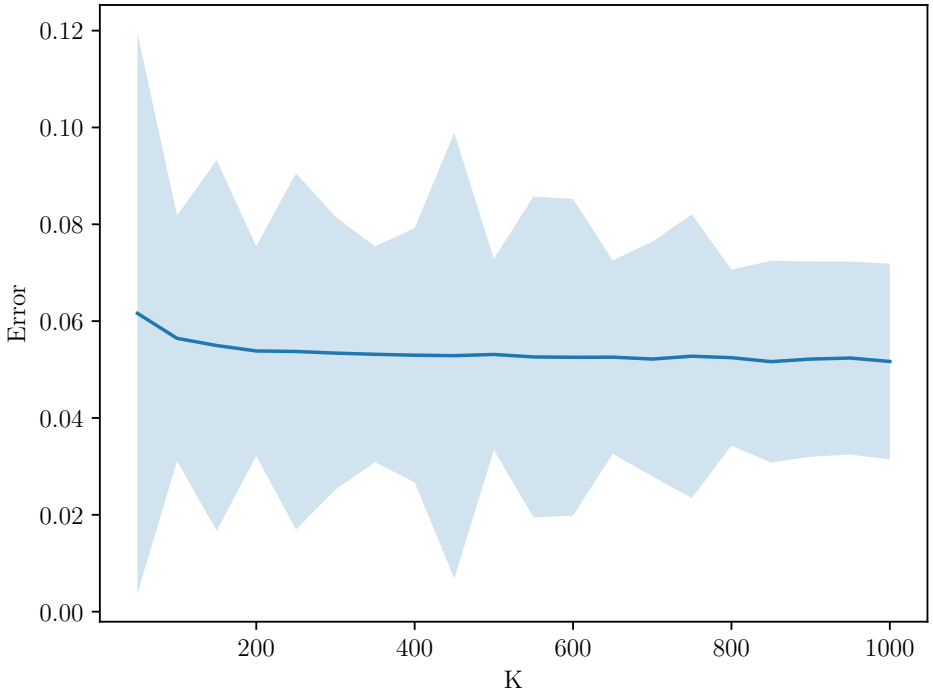


Figure 5: Parameter $K \in [50; 1000]$

Influence of R Parameter R allows us to study error evolution with respect to the number of splitting functions generated during node creation. It is set within the range $[10; 100]$ with increments of 10. Results are provided on Figure 7.

Increasing the number of trials per node usually tends to increase regression quality, yet this happens in a nonlinear fashion. As an example, using either $R = 60$ and $R = 100$ leads to similar results. We can relate this to several factors: first, candidates are drawn using a completely random process, meaning that $R = 60$ and $R = 70$ runs don't share the same 60 first samples. Then, for the very same reason, the pool of candidates P is different between regression models, which can have a

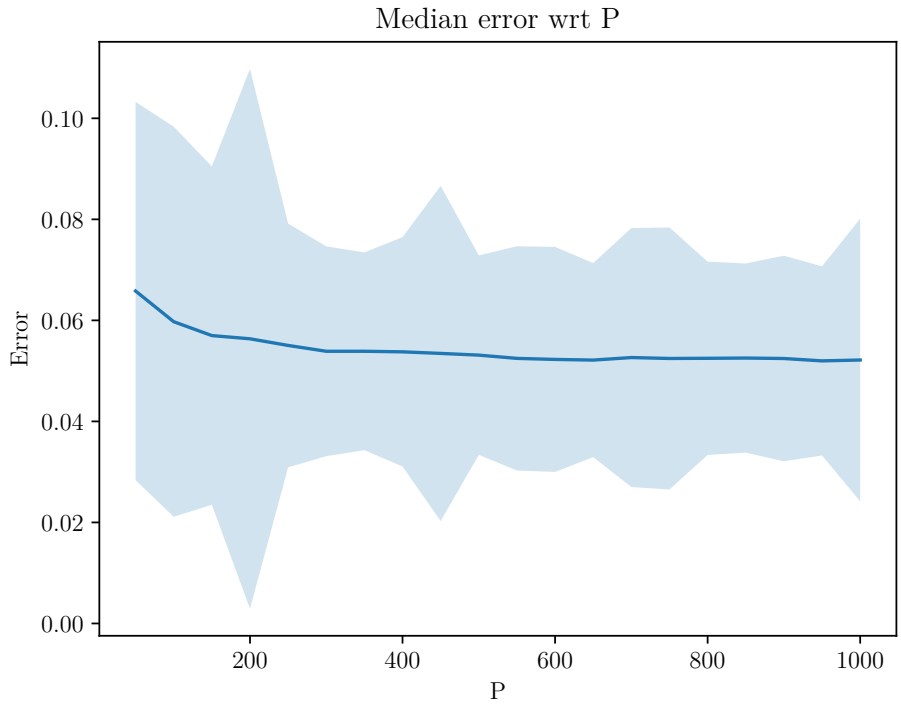


Figure 6: Parameter $P \in [50; 1000]$

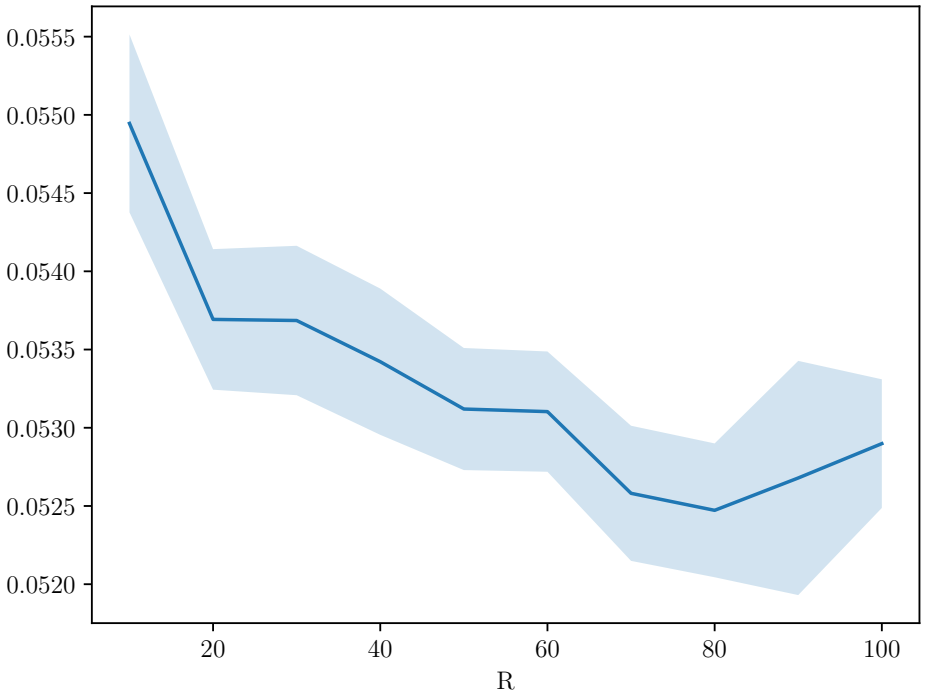


Figure 7: Parameter $R \in [10; 100]$

tremendous impact on results, especially when the size of the pool P is not large enough to generate truly different samples.

Another interesting fact is that increasing the number of candidates R almost always reduce the maximum error of the model. We decide on a trade-off between training speed and accuracy by setting $R = 50$.

Influence of S The correlation between the number of augmented samples per sample S and landmark precision errors can be observed on Figure 8 .

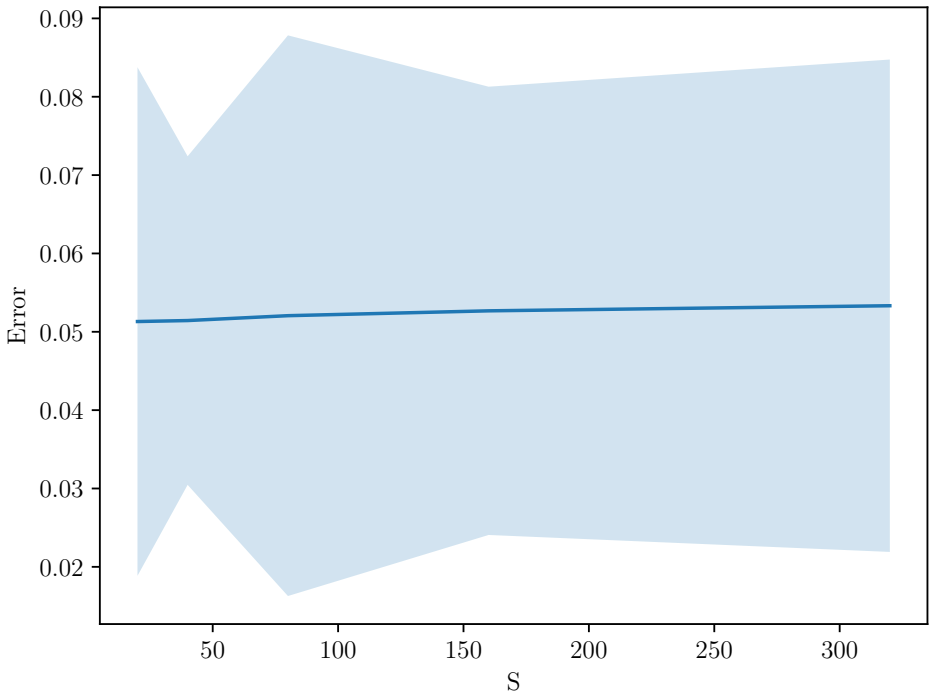
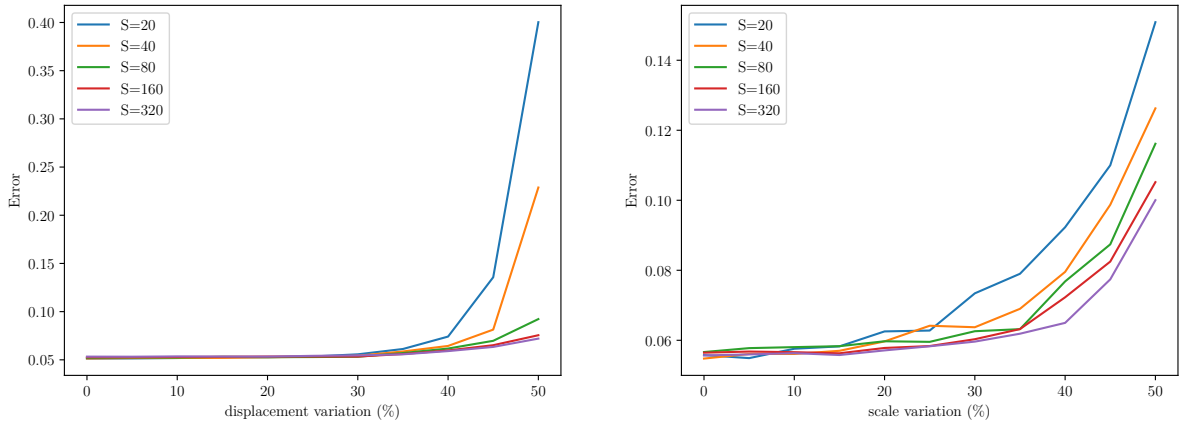


Figure 8: Parameter S

While small values of S seem to be a good choice, we prefer using more augmented samples. The reason is simple: using more samples leads to better robustness against imprecise face detections. Because the bounding boxes used for test data initialization are built using ground truth shapes, the experimental results provided on Figure 8 do not reflect this situation. Instead, this can be revealed by introducing some variations on said bounding boxes. Figures 9a and 9b show landmarking errors using respectively random displacements and random scale variations of ground truth bounding boxes. It appears very clearly that regression-based landmarking is very sensitive to the scale, and even more so on the position of the initial bounding box. Such sensitivity can be significantly reduced using more augmented samples.



(a) Placement noise $\delta_x, \delta_y \in [0\%; 50\%]$

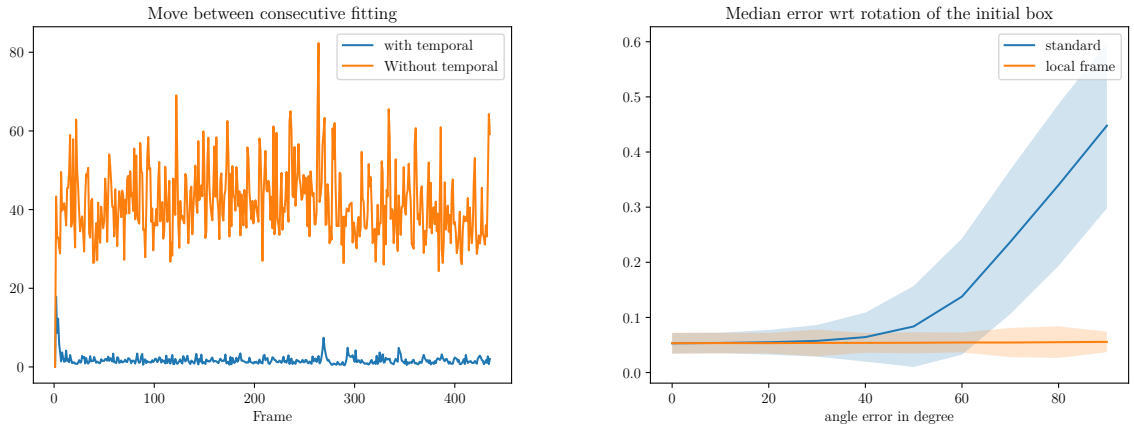
(b) Scale noise $\delta_s \in [0\% ; 50\%]$.

Figure 9: Median landmarking error evolution over parameter variations. Standard deviation appears in transparent blue envelopes.

Note that because of our decision to disturb training data with $\Delta_s = \pm 40\%$ and $\Delta_x, \Delta_y = \pm 50\%$ values, high displacements or scale variations will still increase the landmarking error despite large S values. Like parameter R, the number of augmented samples S only has an impact on training time, so we decided to select a high value of $S = 300$.

6.4 Video tracking strategy

Temporal tracking To test our contribution to temporal tracking improvement explained in section 5, we record a video where the user stays still, using a neutral expression during 15 seconds. In an ideal situation, tracking should either result in a fixed position from frame to frame, or at least be limited to negligible micro-movements of the target face. Unfortunately, the camera sensor introduces noise that may alter fitting results, resulting in an unpleasant noisy result. In Figure 10a we plot the mean displacement of landmarks with standard tracking (orange) and with our method (blue). We can notice that the influence of noise in the input video and results in smoother landmark positions.



(a) (b)

Figure 10: Mean move results in videos, without our contributions (in blue), and with our contributions (in orange). (a) Static content video (b) rotation simulation of the camera.

Robustness to rotation We demonstrate the robustness of our temporal solution using a local frame with a simple example. We simulate a video of a pure in-plane 2D rotation. For each image, we compute a pseudo-video where each image is progressively transformed using a 2D rotation with the origin at the centroid of the face. For all videos we compute the error between ground truth shapes and fitted shapes using two approaches: the standard one using axis aligned initial bounding box and the local orientation frame as described in section 5.2. Figure 10b shows the mean normalized error with both approaches. Note that in the local orientation approach, orientation is computed using the preceding shape fitted and not using the ground truth orientation.

Up to a maximum rotation of 25°, the two approaches led to similar results. But starting from 30°, the local frame approach outperforms the standard one and gives a quite constant error with any rotation angle.

6.5 Comparison with other state-of-the-art methods

This section presents results obtained by our implementation on the 300W competition face data set [1, 2]. 300W consists in 600 in-the-wild images split into two categories: indoor and outdoor. All images are annotated following the Multi-PIE 68 points configuration [35]. For this task, our landmarking algorithm was trained using several public datasets, namely the Face Recognition Grand Challenge (FRGC v.2.0) [36], Labeled Face Parts in the Wild (LFPW) [27], Helen [28], Annotated Faces in the Wild (AFW) [37] and Intelligent Behaviour Understanding Group (iBUG) [2] datasets, totalling 8787 images re-annotated in a consistent manner by a semi-automatic annotation methodology [38].

Because the 300W challenge does not evaluate robustness towards initialization accuracy, we define mean shapes as ground truth bounding boxes like other challengers do [21, 22] and decide not to use the data augmentation strategies described in section 5. We do however define a custom data augmentation strategy for feature point selection called *importance sampling*, where a semantics-driven density distribution is used in place of the classical uniform random function, thus to exploit salient parts of faces as pointed out by Cao *et al.* [12].

Cumulative error curves on the 300W test set are shown on Figure 11. We additionally report the performance of 300W contestants [39, 40, 41, 42, 43, 44], and provide results in terms of area-under-the-curve (AUC) and failure rate (where any fitting with a point-to-point error greater than 0.08 is considered a failure) on Table 1 together with results from 300W contestants as well as the recent Mnemonic Descent Method (MDM) [21]. Both 68 and 51 point errors are provided.

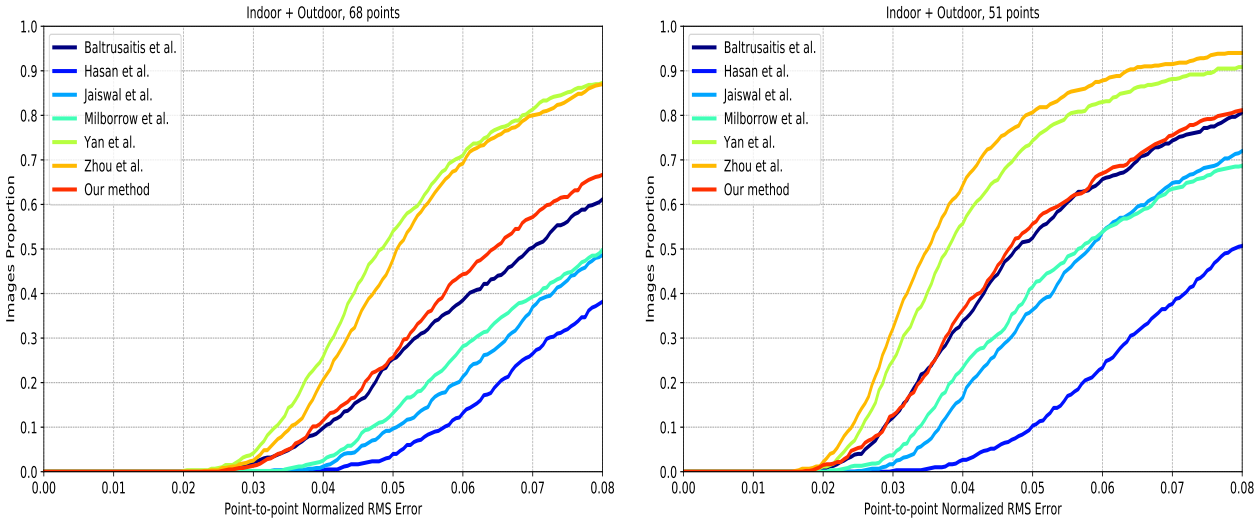


Figure 11: Quantitative results on the test set of the 300W competition (indoor and outdoor combined) for both 68-point (left) and 51-point (right) plots.

	51-points		68-points	
Method	AUC	Failure (%)	AUC	Failure (%)
Milborrow *et al.* [39]	29.89	27.83	15.70	45.50
Jaiswal *et al.* [40]	28.75	25.67	13.84	47.33
Baltrusaitis *et al.* [41]	37.66	17.17	21.94	35.50
Zhou *et al.* [42]	53.30	5.33	36.19	11.00
Hasan *et al.* [43]	14.87	44.50	9.76	55.67
Yan *et al.* [44]	49.13	8.33	38.30	11.17
MDM [21]	56.34	4.20	45.32	6.80
Our method	38.53	17.50	24.40	29.50

Table 1: Quantitative results on the test set of the 300W competition using the AUC (%) and failure rate (calculated at a threshold of 0.08 of the normalized error).

We can notice on Figure 11 that while being outperformed by 300W winners Yan *et al.* and Zhou *et al.*, our method obtains fairly accurate results and competes well with Baltrusaitis *et al.*'s constrained local neural fields-based method. The current tendency for deep learning models to outperform every other method in computer vision tasks makes no exception to face landmarking, as illustrated on Table 1 by MDM results. This is however true solely for accuracy, not computational efficiency. One should keep in mind that deep learning models rely on computationally intensive processes that are incompatible with low-resource, low-power embedded hardware. Bulat *et al.* [22] mention fitting framerate estimations between 28 and 150fps using a NVidia(R) Titan X GPU card, while our method

runs comfortably at 200fps on an Intel(R) Xeon(R) 2.3GHz CPU without much code optimization and no GPU resources whatsoever. Computational costs are probably an issue with second 300W challenge winners Fan *et al.* [45], due to their decision to rely on CNNs, while the other winners Deng *et al.* [46] mention both a strong reliance on face detection accuracy and a testing framerate for each face of 20fps (50ms) using a multi-view, multi-scale and multi-component cascade shape regression strategy.

7 Discussion

Given the results presented in last subsection 6.5, we believe that despite recent advances in deep learning modelling with convolutional kernels, previous works on face landmarking such as cascaded regression trees remain relevant, at least when only low computational resources are available. Trigeorgis *et al.* [21] voice criticism to the fact that binary/tree-based features are not being able to be learnt in an end-to-end manner. We have illustrated the fact that accurate parameter choices have a strong influence on results in the first parts of section 6. Without going back to the drawbacks of purely *hand-crafted* solutions, we strongly believe that clever design strategies such as using semantic face priors for the training procedure, or learning tree or non-linear regression cascades safely in SDM fashion, can lead to even better results in terms of accuracy. As demonstrated early on by Cao [12], CR are able to produce hierarchical facial feature decompositions with induced semantics in a way we find comparable to MDM's natural learning of head pose partitions.

8 Conclusion

In this paper, we have provided an in-depth overview on how to implement and parametrize a cascade of regression tree-based algorithm for face landmarking using state-of-art research publications. All relevant parameters have been explicitly described and their influence over result quality has been investigated thanks to experimental results obtained on well-known face databases. We have also provided benchmark results thanks to the 300W challenge dataset, and shown that despite the current trend of deep learning modelling, CR-based facial landmarking remains fairly accurate, while having the huge advantage of computational efficiency. In an additional step, we contribute to the transition from well-known face landmarking results over *in the wild*, static images, given favourable prior detections, to actual face tracking in real-life video sequences. We show with experimental results that said contributions lead to a smoother tracking and stronger robustness against rotation while reducing and sometimes removing the need for precise face detection beforehand. We noticed that some parameters may have a tremendous impact on robustness and/or precision. For instance, feature selection during the training process is the result of a purely random process. It is very likely that an optimized sampling strategy could increase the outcome of this process. In future work we will study sampling strategies used during regression training so as to improve fitting performance.

Conflicts of interest

The authors state no conflict of interest and have nothing to disclose.

References

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

[2] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, M. Pantic, 300 faces in-the-wild challenge: The first facial landmark localization challenge, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2013, pp. 397–403.

[3] N. Wang, X. Gao, D. Tao, H. Yang, X. Li, Facial feature point detection: A comprehensive survey, Neurocomputing 275 (2018) 50–65.

[4] Y. Wu, Q. Ji, Facial landmark detection: A literature survey, International Journal of Computer Vision 127 (2) (2019) 115–142.

[5] N. Wang, X. Gao, D. Tao, X. Li, Facial feature point detection: A comprehensive survey, CoRR abs/1410.1037 (2014).
URL http://arxiv.org/abs/1410.1037

[6] X. Jin, X. Tan, Face alignment in-the-wild: A survey, CoRR abs/1608.04188 (2016).
URL http://arxiv.org/abs/1608.04188

[7] T. F. Cootes, G. J. Edwards, C. J. Taylor, Active appearance models, in: European conference on computer vision, Springer, 1998, pp. 484–498.

[8] G. Tzimiropoulos, M. Pantic, Gauss-newton deformable part models for face alignment in-the-wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1851–1858.

[9] D. Cristinacce, T. F. Cootes, Feature detection and tracking with constrained local models., in: BMVC, Vol. 1, 2006, p. 3.

[10] J. M. Saragih, S. Lucey, J. F. Cohn, Deformable model fitting by regularized landmark mean-shift, International Journal of Computer Vision 91 (2) (2011) 200–215.

[11] P. Dollár, P. Welinder, P. Perona, Cascaded pose regression, in: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 1078–1085.

[12] X. Cao, Y. Wei, F. Wen, J. Sun, Face alignment by explicit shape regression, International Journal of Computer Vision 107 (2) (2014) 177–190.

[13] V. Kazemi, J. Sullivan, One millisecond face alignment with an ensemble of regression trees, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1867–1874.

[14] S. Ren, X. Cao, Y. Wei, J. Sun, Face alignment at 3000 fps via regressing local binary features, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1685–1692.

[15] E. Sánchez-Lozano, G. Tzimiropoulos, B. Martinez, F. De la Torre, M. Valstar, A functional regression approach to facial landmark tracking, IEEE transactions on pattern analysis and machine intelligence 40 (9) (2017) 2037–2050.

[16] X. Xiong, F. De la Torre, Supervised descent method and its applications to face alignment, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2013, pp. 532–539.

[17] X. Xiong, F. De la Torre, Global supervised descent method, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2664–2673.

[18] L. A. Jeni, J. F. Cohn, T. Kanade, Dense 3d face alignment from 2d videos in real-time, in: 2015 11th IEEE international conference and workshops on automatic face and gesture recognition (FG), Vol. 1, IEEE, 2015, pp. 1–8.

[19] A. Asthana, S. Zafeiriou, S. Cheng, M. Pantic, Incremental face alignment in the wild, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 1859–1866.

[20] A. Jourabloo, M. Ye, X. Liu, L. Ren, Pose-invariant face alignment with a single cnn, in: 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, 2017, pp. 3219–3228.

[21] G. Trigeorgis, P. Snape, M. A. Nicolaou, E. Antonakos, S. Zafeiriou, Mnemonic descent method: A recurrent process applied for end-to-end face alignment, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4177–4187.

[22] A. Bulat, G. Tzimiropoulos, How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks), in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1021–1030.

[23] S. Baker, I. Matthews, Lucas-kanade 20 years on: A unifying framework, International journal of computer vision 56 (3) (2004) 221–255.

[24] I. Matthews, S. Baker, Active appearance models revisited, International journal of computer vision 60 (2) (2004) 135–164.

[25] L. Breiman, Random forests, Machine learning 45 (1) (2001) 5–32.

[26] P. Viola, M. J. Jones, Robust real-time face detection, Int. J. Comput. Vision 57 (2) (2004) 137–154. doi:10.1023/B:VISI.0000013087.49260.fb.
URL http://dx.doi.org/10.1023/B:VISI.0000013087.49260.fb

[27] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, N. Kumar, Localizing parts of faces using a consensus of exemplars, IEEE transactions on pattern analysis and machine intelligence 35 (12) (2013) 2930–2940.

[28] V. Le, J. Brandt, Z. Lin, L. Bourdev, T. S. Huang, Interactive facial feature localization, in: European Conference on Computer Vision, Springer, 2012, pp. 679–692.

[29] X. P. Burgos-Artizzu, P. Perona, P. Dollár, Robust face landmark estimation under occlusion, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 1513–1520.

[30] C. Cao, Q. Hou, K. Zhou, Displaced dynamic expression regression for real-time facial tracking and animation, ACM Transactions on Graphics (TOG) 33 (4) (2014) 43.

[31] H. Yang, R. Zhang, P. Robinson, Human and sheep facial landmarks localisation by triplet interpolated features, CoRR abs/1509.04954 (2015).
URL http://arxiv.org/abs/1509.04954

[32] H. Yang, X. Jia, C. C. Loy, P. Robinson, An empirical study of recent face alignment methods, CoRR abs/1511.05049 (2015).
URL http://arxiv.org/abs/1511.05049

[33] G. B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled faces in the wild: A database for studying face recognition in unconstrained environments, Tech. rep., Technical Report 07-49, University of Massachusetts, Amherst (2007).

[34] C. Cao, Y. Weng, S. Zhou, Y. Tong, K. Zhou, Facewarehouse: A 3d facial expression database for visual computing, IEEE Transactions on Visualization and Computer Graphics 20 (3) (2014) 413–425.

[35] R. Gross, I. Matthews, J. Cohn, T. Kanade, S. Baker, Multi-pie, Image and Vision Computing 28 (5) (2010) 807–813.

[36] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, W. Worek, Overview of the face recognition grand challenge, in: Computer vision and pattern recognition, 2005. CVPR 2005. IEEE computer society conference on, Vol. 1, IEEE, 2005, pp. 947–954.

[37] X. Zhu, D. Ramanan, Face detection, pose estimation, and landmark localization in the wild, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 2879–2886.

[38] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, M. Pantic, A semi-automatic methodology for facial landmark annotation, in: Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on, IEEE, 2013, pp. 896–903.

[39] S. Milborrow, T. Bishop, F. Nicolls, Multiview active shape models with sift descriptors for the 300-w face landmark challenge, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2013, pp. 378–385.

[40] S. Jaiswal, T. Almaev, M. Valstar, Guided unsupervised learning of mode specific models for facial point detection in the wild, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2013, pp. 370–377.

[41] T. Baltrusaitis, P. Robinson, L.-P. Morency, Constrained local neural fields for robust facial landmark detection in the wild, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2013, pp. 354–361.

[42] E. Zhou, H. Fan, Z. Cao, Y. Jiang, Q. Yin, Extensive facial landmark localization with coarse-to-fine convolutional network cascade, in: Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on, IEEE, 2013, pp. 386–391.

[43] M. Hasan, C. Pal, S. Moalem, Localizing facial keypoints with global descriptor search, neighbour alignment and locally linear models, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2013, pp. 362–369.

[44] J. Yan, Z. Lei, D. Yi, S. Li, Learn to combine multiple hypotheses for accurate face alignment, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2013, pp. 392–396.

[45] H. Fan, E. Zhou, Approaching human level facial landmark localization by deep learning, Image and Vision Computing 47 (2016) 27–35.

[46] J. Deng, Q. Liu, J. Yang, D. Tao, M3 csr: Multi-view, multi-scale and multi-component cascade shape regression, Image and Vision Computing 47 (2016) 19–26.