Confidence-Weighted Local Expression Predictions for Occlusion Handling in Expression Recognition and Action Unit detection

Arnaud Dapogny, Kevin Bailly, Séverine Dubuisson

Confidence-Weighted Local Expression Predictions for Occlusion Handling in Expression Recognition and Action Unit detection

Arnaud Dapogny[1]
arnaud.dapogny@isir.upmc.fr

Kevin Bailly[1]
kevin.bailly@isir.upmc.fr

Séverine Dubuisson[1]
severine.dubuisson@isir.upmc.fr

[1] Sorbonne Universités, UPMC Univ Paris 06, CNRS, ISIR UMR 7222
4 place Jussieu 75005 Paris

Abstract

Fully-Automatic Facial Expression Recognition (FER) from still images is a challenging task as it involves handling large interpersonal morphological differences, and as partial occlusions can occasionally happen. Furthermore, labelling expressions is a time-consuming process that is prone to subjectivity, thus the variability may not be fully covered by the training data. In this work, we propose to train Random Forests upon spatially defined local subspaces of the face. The output local predictions form a categorical expression-driven high-level representation that we call Local Expression Predictions (LEPs). LEPs can be combined to describe categorical facial expressions as well as Action Units (AUs). Furthermore, LEPs can be weighted by confidence scores provided by an autoencoder network. Such network is trained to locally capture the manifold of the non-occluded training data in a hierarchical way. Extensive experiments show that the proposed LEP representation yields high descriptive power for categorical expressions and AU occurrence prediction, and leads to interesting perspectives towards the design of occlusion-robust and confidence-aware FER systems.

Introduction

Automatic Facial Expression Recognition (FER) from still images is an ongoing research field which is key to many human-computer applications, such as consumer robotics or social monitoring. To address these problems, a lot of emphasis has been put by the psychological community in order to define models that are both accurate and exhaustive enough to describe facial expressions.

Perhaps one of the most long-standing model for describing the expressions is the discrete categorization proposed by Paul Ekman within his cross-cultural studies [1], in which he introduced six universally recognized basic expressions (*happiness*, *anger*, *sadness*, *fear*, *disgust* and *surprise*). Along with a *neutral* state, this has been used as an underlying expression model for most attempts at developing a prototypical expression recognition system [2], [3]. However, this model faces limitations for dealing with spontaneous facial expressions [4], as many of our daily affective behaviors may not be translated in terms of prototypical emotions. Nevertheless, the annotation process is rather intuitive, thus there exists a large corpus of labelled data.

Another approach is the continuous affect representation [5] that consists in projecting expressions onto a restricted number of latent dimensions. A popular example of such model is the valence/activation (relaxed *vs.* aroused)/power (feeling of control)/expectancy (anticipation) model. It is often simplified as a two-dimensional valence-activation representation. However, using such a low-dimensional embedding of facial expressions can cause the loss of information. Indeed, expressions such as *surprise* are not represented correctly whereas others can not be separated well (*fear vs. anger*). Last but not least, the annotation process is less intuitive than with the categorical representation.

Finally, an alternative facial expression model is the Facial Action Coding System (FACS) [6]. It consists in describing facial expressions as a combination of 44 facial muscle activations that are refered to as Action Units (AUs). AUs is a face representation that may be less subject to interpretation. It can theoretically be used in accordance with the so-called Emotional FACS (EMFACS) rules in order to describe a broader range of spontaneous expressions. However, the main drawback of the FACS-coding approach is that the annotation tends to be a time-consuming process. Furthermore, FACS coders have to be highly trained, hence limiting the quantity of available data.

In the meantime, as stated in [7], FER from still images is a challenging task as there may exist large variability in the morphology or in the expressiveness of different persons. Furthermore, countless configurations of partial occlusion can occasionally happen (e.g. with hand or accessories). As a result, this variability cannot be fully covered by restricted amounts of available training data. For those reasons, this paper introduces a new categorical expression-driven representation that we call Local Expression Predictions (LEPs). LEPs can be learned efficiently on the available expression datasets, and can serve multiple purposes such as occlusion handling for (global) categorical expression recognition, as well as confidence-aware AU detection.

1 Related work

In this section we review recent approaches covering FER, with an emphasis on methods addressing the problem of partial occlusions. We also describe methods for AU detection.

1.1 Occlusion handling in categorical FER

Most recent approaches covering FER from still images work in controlled conditions, on a frontal view and lab-recorded environments [2, 3]. Shan *et al.* [8] evaluated the recognition accuracy of Local Binary Pattern (LBP) features. Zhong *et al.* [9] proposed to learn active facial patches that are relevant for FER. Zhao *et al.* [10] designed a unified multitask framework for simultaneously performing facial alignment, head pose estimation and FER. Such approaches showed satisfying results in constrained scenarios, but they can face difficulties on more challenging benchmarks [11].

Eleftheriadis *et al.* [12] used disciminative shared Gaussian processes to perform pose-invariant FER. Liu *et al.* [13] introduced a deep neural network that learns local features relevant for Action Unit prediction, and use it as an intermediate representation for categorical FER. The authors also studied the use of unlabelled data [14] to regularize the network training, further enhancing its predictive capacities for FER in the wild. However, none of these approaches explicitly addresses the problem of facial occlusions that are likely to happen in such unconstrained cases.

Kotsia *et al.* [15] studied the impact of human perception of facial expressions under partial occlusions, and the predictive capacities of automated systems thereof. Cotter *et al.* [16] used sparse decomposition to perform FER on corrupted images. Ghiasi *et al.* [17] use a discriminative approach for facial feature point alignment under partial occlusions. Those approaches rely on explicitly incorporating synthetic occluded data in the training process, and thus struggle to deal with realistic, unpredicted occluding patterns. Zhang *et al.* [18] trained classifiers upon random Gabor-based templates. They evaluated their algorithms on synthetically occluded face images, showing that their approach leads to a better recognition rate when the same occluded examples are used for training and testing. Should this not be the case, unpredicted mouth/eye occlusions still lead to a significant loss of performance. Huang *et al.* [19] proposed to automatically detect the occluded regions using sparse decomposition residuals. However, the proposed approach may not be flexible enough, as the occlusion detection only outputs binary decisions, and as the face is explicitly divided into only three subparts (eyes, nose and mouth). This limits the capacities of the method to deal with unpredicted forms of occlusion. Finally, another approach consists in learning generative models of non-occluded faces, as it was done by Ranzato *et al.* [20]. When testing on a partially occluded face image, the occluded parts can be generated back and expression recognition can be performed. The pitfall of such an approach is that training can be computationally expensive and does not allow the use of heterogeneous features (e.g. geometric/appearance descriptors).

1.2 Action Unit detection

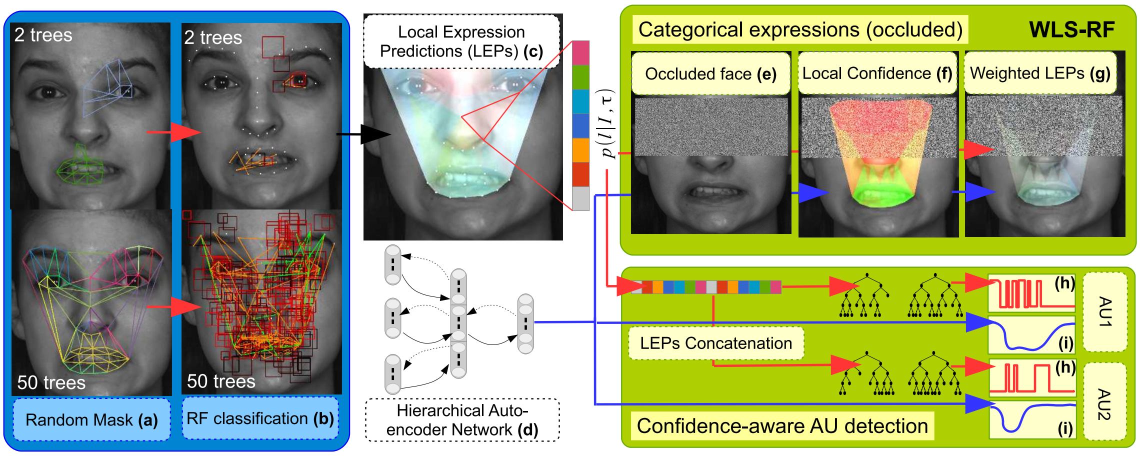

Figure 1: LEPs and applications to categorical expression recognition, occlusion handling in FER and AU detection. Randomized trees are trained upon local subspaces generated under the form of random facial masks **(a)**, on which binary feature candidates are generated and selected **(b)**. The local predictions outputted by the trees can be aggregated into categorical expression-driven high-level LEP representations **(c)**. Given an occluded face image **(e)**, an occlusion-robust categorical expression prediction can be outputted by weighting LEPs with confidence scores **(f)** given by a hierarchical autoencoder network **(d)**. Furthermore, LEP features can be used to predict AU occurrence **(h)**, for which an AU-specific confidence measurement can be provided **(i)**. Best viewed in color.

AU detection is traditionally performed by applying binary classification upon high-dimensional, low-level image descriptors such as LBP or Local Phase Quantification (LPQ) features [21]. Senechal *et al.* proposes to embed heterogeneous (geometric/appearance) features within a multi-kernel SVM framework [22]. These features can be extended to spatio-temporal volumes with the Three Orthogonal Planes (TOP) paradigm, as proposed in [4] for LBP-TOP and [21] for LPQ-TOP. In the meantime, Chu *et al.* [23] introduced a new learning algorithm that personalizes a generic classification framework by attenuating person-specific biases. Nicolle *et al.* [24] proposed an AU intensity prediction by a novel multi-task formulation of the metric learning for kernel regression method. However, there seems to be a gap between low-level feature descriptors (e.g. LBP, LPQ, SIFT) and high-end learning algorithms applied to AU detection, that could be filled by learning representations from a large corpus of categorical expression-labelled examples.

Recently, Ruiz *et al.* [25] introduced a new framework where categorical expressions are learnt from prior knowledge between this visible task and a set of hidden tasks that correspond to AU detection predictions. Thus, they exploit relationships between those two tasks to learn AU detectors with (SHTL) or without (HTL) FACS-labelled training data, by explicitly combining AUs into categorical expressions *a la* EMFACS. Conversely, we propose to describe AUs as a combination of Local Expression Predictions (LEPs). Furthermore, to the best of our knowledge, this is the first time that a confidence assessment is provided for AU detection.

2 Method Overview

In this work, we introduce a new Local Expression Prediction (LEP) representation that can be learned from data labelled with categorical expressions, as described in Figure 1. During training, local subspaces are generated under the form of random facial masks (a), onto which binary candidate features can be selected (b) to train randomized trees. The local LEPs (c) outputted by local subspace Random Forests can be used for multiple purposes that are depicted below. Furthermore, we also introduce a hierarchical autoencoder network (d), which can be used to capture the local manifold of non-occluded faces around separate aligned feature points. When applied on a potentially occluded face image (e), the reconstruction error outputted by such a network provides a confidence measurement of how close a face region lies from the training data manifold (f), with high and low confidences depicted in green and red respectively. This local confidence measurement can be used to weight LEPs (g) in order to provide an occlusion-robust expression prediction (WLS-RF). Finally, LEPs can be used to predict AU occurrence (h). Once again, the autoencoder network can be used to provide AU-specific confidence measurements (i). Our contributions are thus the following:

(1) A method for training random trees upon spatially-defined local subspaces, which consists in generating random masks covering a specified fraction of the face. These local trees can be combined to produce high-level expression-driven representations that we refer to as LEPs, which can be used for categorical FER or AU prediction.

(2) A hierarchical autoencoder network for learning local non-occluded face manifolds, which can be used to provide local confidence measurements.

(3) The confidence measurements outputted by the autoencoder network can be used to enhance robustness to occlusions for categorical FER, as well as to assess confidence for AU detection.

The rest of the paper is organized as follows: in Section 3 we discuss the proposed autoencoder network architecture (Section 3.1) and how it is trained to capture the local manifold around facial feature points (Section 3.2). Section 4 describes how we learn the Local Expression Predictions via local subspace random forests (Section 4.1) with heterogeneous binary feature candidates (Section 4.2). In particular, we explain in Section 4.3 how those local representations can be effectively combined and weighted to produce occlusion-robust predictions, and in Section 4.4 how LEPs can be used for confidence assessment in AU detection. Finally, in Section 5 we show that our approach significantly improves the state-of-the-art for categorical FER on multiple datasets (described in Section 5.1), both on the non-occluded (Section 5.3.1) and occluded cases (Section 5.3.2). We also demonstrate in Section 5.4 the interest of our LEP representation for AU activation prediction and the relevance of the AU-specific confidence measurement. Finally, Section 6 provides a conclusion as well as a few perspectives raised in the paper.

3 Manifold Learning of non-occluded faces with a hierarchical autoencoder network

Given a number of aligned facial feature points that can be provided by an off-the-shelf alignment algorithm (such as the SDM tracker [26]), we use an autoencoder network to model the local face pattern manifold. This network will thus be used to provide a local confidence measurement that is used to weight LEPs for occlusion-robust FER.

3.1 Network architecture

Autoencoders are a particular type of neural network that can be used for manifold learning. Compared with other approaches such as PCA [27], autoencoders offer the advantage to theoretically be able to model complex manifolds using non-linear encoding and regularization criteria, such as denoising [28] or contractive penalties [29]. As compared to manifold forests [30], autoencoders can be trained on high-dimensional features without falling into the pitfall of low-rank deficiency. Furthermore, they benefit from an efficient training using stochastic gradient descent, as well as the possibility of online fine-tuning for subject-specific calibration.

As shown in Figure 2, we use a 2-layer architecture. First, we extract Histograms of Oriented Gradients (HOG) within the neighbourhood of each feature point aligned on the face image $\mathcal{I}$. The choice of learning a manifold of HOG patterns rather than gray levels comes from the fact that HOG are used for both facial alignment and the LEP generation pipeline. Thus, the reconstruction error of these patterns provides a confidence measurement that is relevant for both tasks. In order to ensure fast HOG extraction, we use integral feature channels as introduced in [31]. Horizontal and vertical gradients are computed on the image and used to generate 9 feature maps. The first of these contains the gradient magnitude, and the 8 remaining correspond to a 8-bin quantization of the gradient orientation. Then, integral images are computed from these feature maps to output the 9 feature channels. Also, storing the gradient magnitude within the first channel allows to normalize the histograms as in standard HOG implementations. Thus, HOG features can be computed very efficiently by using only 4 access to previously stored integral channels (plus normalization). Note that those feature channels can be computed only once for the three steps of the pipeline.

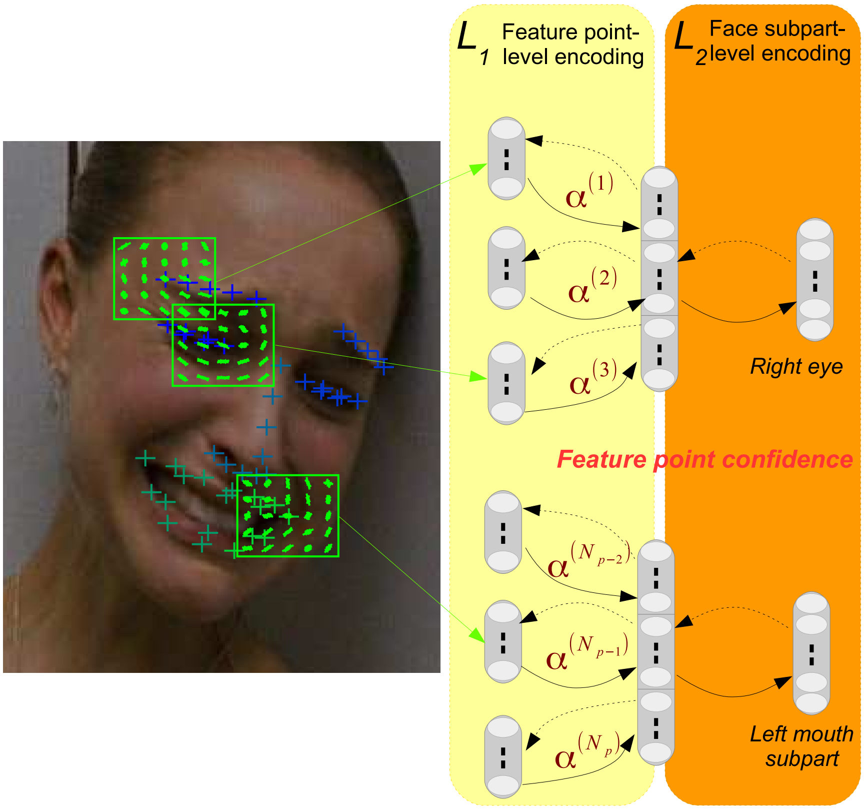

Figure 2: Architecture of our hierarchical autoencoder network. The network is composed of 2 layers: the first one (L_1) captures the texture variations (HOG descriptors) around the separate aligned feature points. The second one (L_2) is defined over 5 face subparts, each of which embraces multiple points whose appearance variations are closely related. The network outputs a confidence score $\alpha^{(k)}$ for each of the N_p feature points.

The local descriptor $\mathbf{\Psi}^{(k)}$ for a specific feature point k consists in the concatenation of gradient magnitudes and quantized orientation values in 5×5 cells around this feature point, with a total window size equal to a third of the inter-ocular distance. This descriptor of dimension 225 then feeds the N_p autoencoders (one per feature point) of the first layer (L_1) which are trained to reconstruct non-occluded patterns. Because occlusion of local patterns extracted at the feature point level are not independent (*i.e.* a feature point close to an occluded area is more likely to be occluded itself), we employ a second layer (L_2) of autoencoders, that are trained to reconstruct non-occluded patterns of groups of encoded feature point descriptors. Those groups represent five face subparts (left and right eyes, nose, left and right parts of the mouth) from which the local patterns are closely related. Specifically, L_1 is composed of 125 units for each landmark. L_2 layer for a feature point group contains $65 \times N$ units ($\frac{1}{2}$ compression), where N is 12,12,8,11 and 11 respectively for left/right eye, nose and left/right mouth areas.

3.2 Training the network

Autoencoders are generally trained in an unsupervised way, one layer at a time, by optimizing a reconstruction criterion. The input descriptor $\mathbf{\Psi}^{(k)}$ at feature point k is first encoded via the L_1 encoding layer into an intermediate representation $\mathbf{y}^{(k)} = h^1(\mathbf{\Psi}^{(k)})$:

$$\mathbf{y}^{(k)} = \sigma\left(\boldsymbol{w}^{(k)}.\mathbf{\Psi}^{(k)} + \mathbf{b}^{(k)}\right) \tag{1}$$

Where σ is a sigmoid function, $\boldsymbol{w}^{(k)}$ and $\mathbf{b}^{(k)}$ are respectively the neuron weight matrix and bias vector of the L_1 neuron layer for feature point k. The output is then typically computed as the input reconstruction $\tilde{\boldsymbol{\Psi}}^{(k)} = g^1(\mathbf{y}^{(k)})$ using an affine decoder with tied input weights to reduce the number of parameters:

$$\tilde{\boldsymbol{\Psi}}^{(k)} = \boldsymbol{w}^{(k)T}.\mathbf{y}^{(k)} + \mathbf{c}^{(k)} \tag{2}$$

Where $c^{(k)}$ is the decoder bias vector. Then the set of K encoded descriptors $\{\tilde{\boldsymbol{\Psi}}^{(k)}\}_{k=1...K}$ associated to feature points $k = 1...K$ that belong to the face subpart m are concatenated to form the input $\xi^{(m)}$ of the layer L_2 for that subpart. Once again, the input of the L_2 layer is successively encoded into a intermediate representation $\mathbf{z}^{(m)} = h^2(\xi^{(m)})$ and decoded in the same way into a reconstructed version $\tilde{\xi}^{(m)} = g^2(\mathbf{z}^{(m)})$:

$$\mathbf{z}^{(m)} = \sigma\left(\boldsymbol{w'}^{(m)}.\xi^{(m)} + \mathbf{b'}^{(m)}\right) \tag{3}$$

$$\tilde{\xi}^{(m)} = \boldsymbol{w'}^{(m)T}.\mathbf{z}^{(m)} + \mathbf{c'}^{(m)} \tag{4}$$

Thus, each layer is trained separately using stochastic gradient descent and backpropagation. More specifically, the input descriptors for each layer are presented sequentially. For example, a forward pass through the L_1 layer provides a reconstructed version $\tilde{\boldsymbol{\Psi}}^{(k)}$ of $\boldsymbol{\Psi}^{(k)}$. The squared $\mathcal{L}_2$-loss is then computed and weighted by a learning rate parameter to provide the parameter update $(\delta\boldsymbol{w}^{(k)}, \delta\mathbf{b}^{(k)}, \delta\mathbf{c}^{(k)})$. We tried various combinations of training parameters and the best reconstruction results were obtained by applying 15000 stochastic gradient updates with alternating sampling between the expression classes in the databases. Indeed, we want the network to be able to reconstruct local variations of all possible expressive patterns on an equal foot. We also use a constant learning rate of 0.01 as well as a weight decay of 0.001, which seems to provide good results in testing. Finally, we found that adding 25% randomly generated masking noise provided satisfying results. From a manifold learning perspective, the goal of using such denoising criterion is to learn to project corrupted examples (e.g. partially occluded ones, which lie further from the manifold) back on the training data manifold. Such example will be reconstructed closer to the training data and its confidence shall be smaller.

3.3 Local confidence measurement

Given a face image $\mathcal{I}$, we define the point-wise confidence $\alpha^{(k)}(\mathcal{I})$ for point k as the $\mathcal{L}_2$-loss (*i.e.* the reconstruction error) between the HOG descriptor $\boldsymbol{\Psi}(\mathcal{I})$ extracted from this feature point, and its reconstruction $\tilde{\boldsymbol{\Psi}}$ outputted by the network, after successively encoding by layers L_1 then L_2, and decoding in the opposite order. By abuse of notation, we have:

$$\alpha^{(k)}(\mathcal{I}) = ||\boldsymbol{\Psi}^{(k)} - g^1 \circ g^2 \circ h^2 \circ h^1(\boldsymbol{\Psi}^{(k)})||^2 \tag{5}$$

The choice of using an Euclidean distance as a confidence score seems natural as it is optimized during training. We also introduce a confidence measurement $\alpha^{(\tau)}(\mathcal{I})$ defined over triangles $\tau = \{k_1, k_2, k_3\}$ of the facial mesh as:

$$\alpha^{(\tau)}(\mathcal{I}) = \min(\alpha^{(k_1)}(\mathcal{I}), \alpha^{(k_2)}(\mathcal{I}), \alpha^{(k_3)}(\mathcal{I})) \tag{6}$$

As highlighted in the following experiments, this triangle-wise confidence measurement can thus be used to weight LEPs to enhance the robustness to partial occlusions to a significant extent.

4 Local Expression Predictions

4.1 Learning local trees with random facial masks

Random Forests (RF) is a popular learning framework introduced in the seminal work of Breiman [32]. They have been used to a significant extent in computer vision, and for FER tasks in particular [10, 33], due to their ability to nicely handle high-dimensional data such as images as well as being naturally suited for multiclass classification tasks.

In the classical RF framework, each tree of the forest is grown using a subset of training examples (bagging) and a subspace of the input dimension (random subspace). Individual trees are then grown using a greedy procedure that involves, for each node, the generation of a number of binary split candidates that consist in features associated with a threshold. Each candidate thus defines a partition of the labelled training data. The “best" binary feature is chosen among all features as the one that minimizes an impurity criterion (which is generally defined as either the Shannon entropy or the Gini impurity). Then, the above steps are recursively applied for the left and right subtrees with accordingly rooted data until the label distribution at each node becomes homogeneous, where a leaf node can be set. As stated in [32], the rationale behind training each tree on a random subspace of the input dimension is that the prediction accuracy of the whole forest depends on both the strength of individual trees and on the independence of the predictions. Thus, by growing individually weaker (e.g. as compared to C4.5) but more decorrelated trees, we can combine these into a more accurate tree collection.

Following this idea, we propose an adaptation of the RF framework that uses spatially-defined Local Subspaces (LS) instead of the traditional Random Subspaces (RS). Each tree is trained using a restricted subspace corresponding to a specific part of the face. The aggregation of local models gives rise to local representations that we call Local Expression Predictions (LEPs). Note that this is not the first time that the output classification predictions of RFs are used as features for a subsequent task. For instance, Ren *et al.* [34] used local binary features to construct a cascaded feature point alignment method. However, contrary to [34], we construct our LEP representation by locally averaging predictions and not by directly using the output prediction of the trees. Furthermore, LEPs offer several advantages over using a set of trees defined on the whole face:

(1) LEPs can be aggregated to provide categorical FER. Those local models (LS-RF) can theoretically capture more diverse information compared to a global one by “forcing” the trees to use less informative features, that can still hold some predictive power.

(2) We can use the confidence outputted by the autoencoder network in Section 3 to weight the LEPs for which the pattern lies further from the training data manifold (WLS-RF). For example, in case of occlusion or drastic illumination changes, we can still use the information from the other face subparts to predict the expression.

(3) LEPs can be used as an intermediate representation for the task of describing Action Units (AUs). Noteworthy, AU classification could benefit from LEPs trained on larger corpus labelled with categorical expressions, as annotation is less time-consuming than FACS coding.

The local trees are trained using Algorithm 1. For each tree t in the forest, we generate a face mask M_t defined over triangles τ on facial feature points of a precomputed mean shape $\bar{f}$. The mask is initialized with a single triangle randomly selected from the mesh. Then, neighbouring triangles are added until the total surface covered by the selected triangles w.r.t. $\bar{f}$ becomes superior to hyperparameter R, that represents the (approximate) surface that should be covered by each tree. Finally, tree t is grown on the subspace that corresponds to the facial mask M_t.

4.2 Candidate feature generation

We use a combination of geometric (*i.e.* computed from aligned facial feature points) and appearance features $\phi^{(1)}, \phi^{(2)}, \phi^{(3)}$, as in [33]. Each of these features have different parameters that are generated on-the-fly during training by uniform sampling over their respective variation range.

We use two different geometric feature templates which are generated from the set of N_p facial feature points $f(\mathcal{I})$, provided by an off-the-shelf facial alignment algorithm [26]. The first of these templates is the Euclidean distance between feature points f_a and f_b, normalized w.r.t. intra-ocular distance $iod(f)$ for scale invariance (Equation 7).

$$\phi^{(1)}_{a,b}(\mathcal{I}) = \frac{||f_a - f_b||_2}{iod(f)} \tag{7}$$

Because the feature point orientation is discarded in feature $\phi^{(1)}$ we use the angles between feature points f_a, f_b and f_c as our second geometric feature $\phi^{(2)}_{a,b,c,\lambda}$. In order to ensure continuity for angles around 0, $\phi^{(2)}$ outputs either the cosine or sine of angle $\widehat{f_a f_b f_c}$, depending on the value of the boolean parameter λ (Equation (8)):

Algorithm 1 Training Local Subspace Random Forest

```
input: images $\mathcal{I}$ with labels $l$ and feature points $f(\mathcal{I})$
  compute $\bar{f}$, the mean shape
  compute $s(\tau(\bar{f}))$, surface of triangles $\tau$ on mean shape
  for $t = 1$ to $T$ do
      randomly select a triangle $\tau_i$
      $r \leftarrow s(\tau_i)$
      initialize mask $M_t \leftarrow \{\tau_i\}$
      while $r < R$ do
          draw a list of candidate neighbouring triangles
          randomly select a triangle $\tau_j$ from that list
          $r \leftarrow r + s(\tau_j)$
          $M_t \leftarrow M_t \cup \{\tau_j\}$
      end while
      randomly select a fraction $\tilde{\mathcal{S}}_t \subset \mathcal{S}$ of subjects
      balance bootstrap $\tilde{\mathcal{S}}_t$ with downsampling
      grow tree $t$ on bootstrap $\tilde{\mathcal{S}}_t$ and input subspace $M_t$
  end for
  output: tree predictors $p_t(l|\mathcal{I})$ with associated masks $M_t$
```

$$\phi^{(2)}_{a,b,c,\lambda}(\mathcal{I}) = \lambda \cos(\widehat{f_a f_b f_c}) + (1-\lambda)\sin(\widehat{f_a f_b f_c}) \tag{8}$$

As appearance features, we use HOGs for their descriptive power and robustness to global illumination changes. Integral HOG feature channels were already computed in Section 3.1 for confidence weight computation. Thus, we define feature template $\phi^{(3)}_{\tau,ch,sz,\alpha,\beta,\gamma}$ as an integral histogram computed over channel ch within a window of size sz normalized w.r.t. the intra-ocular distance. Such histogram is evaluated at a point defined by its barycentric coordinates α, β and γ w.r.t. vertices of a triangle τ defined over feature points $f(\mathcal{I})$.

Each of these candidate features are associated with a set of thresholds θ to produce binary split candidates. In particular, for each feature template $\phi^{(i)}$, the upper and lower bounds are estimated from training data beforehand and candidate thresholds are drawn from a uniform distribution in the range of these values.

4.3 Occlusion-robust expression recognition

When testing, a face image $\mathcal{I}$ is successively rooted left or right for each tree t depending of the outputs of the binary tests stored in the tree nodes, until it reaches a leaf. The tree t thus outputs a probability vector $p_t(l|\mathcal{I})$ whose components are either 1 for the represented class, or 0 otherwise. Prediction probabilities are then averaged among the T trees of the forest (Equation (9)).

$$p(l|\mathcal{I}) = \frac{1}{T}\sum_{t=1}^{T} p_t(l|\mathcal{I}) \tag{9}$$

Those prediction probabilities are computed similarly for the global RF (RS-RF) and the LS-RF. However, for LS-RF the output probabilities of the trees have some degrees of locality and we can write the above formula as a sum over local probabilities defined for each triangle (Equation (10)).

$$p(l|\mathcal{I}) = \frac{1}{T}\sum_{\tau} Z_\tau p(l|\mathcal{I},\tau) \tag{10}$$

Where $p(l|\mathcal{I},\tau)$ is the Local Expression Prediction (LEP) probability vector associated with triangle τ on the facial mesh:

$$p(l|\mathcal{I},\tau) = \frac{1}{Z_\tau}\sum_{t=1}^{T} \frac{\delta(\tau \in M_t) p_t(l|\mathcal{I})}{|M_t|} \tag{11}$$

With $\delta(\tau \in M_t)$ being a function that returns 1 if triangle τ belongs to mask M_t, and 0 otherwise. $|M_t|$ is the number of times tree t is used in Equation (10), and Z_τ is the sum of prediction values for all expression classes l. Thus, a global expression probability is defined by a (normalized) sum of LEPs. Note that those LEP vectors $p(l|\mathcal{I}, \tau)$ are not strictly limited to triangle τ but defined within its neighbourhood, with a radius that depends on hyperparameter R. The setting of R thus controls the locality of the trees, as it will be discussed in the experiments.

Moreover, LEPs can be weighted by local confidence measurements to give rise to the Weighted Local Subspace Random Forest model (WLS-RF):

$$p(l|\mathcal{I}) = \frac{\sum\limits_{\tau} \alpha^{(\tau)} Z_\tau p(l|\mathcal{I}, \tau)}{\sum\limits_{\tau} \alpha^{(\tau)} Z_\tau} \tag{12}$$

Where $\alpha^{(\tau)}$ is the triangle-wise confidence measurement that is outputted by the autoencoder network described in Section 3, for triangle τ. This weighting scheme allows to better handle partial occlusions, by downweighting the local RFs associated with the most unreliable appearance patterns.

4.4 Action Unit detection

4.4.1 From LEPs to Action Units

LEPs are local responses related to categorical facial expressions. Thus, it makes sense to assume that LEPs can somehow be related to AUs and constitute a good high-level representation for AU recognition. To this end, Figure 3 describes the AU recognition framework, in which LEP vectors corresponding to each triangle are extracted by a first layer of local trees, trained on a categorical expression dataset. The concatenation of all LEP vectors $p(l|\mathcal{I}, \tau)$ for every expression l (6 universal expressions plus the neutral one) and triangle τ of the facial mesh gives rise to a $7 \times N_\tau$ feature vector used by a second layer of trees defined for each AU (with N_τ the number of triangles of the facial mesh). Thus, the AU recognition layer is trained on a FACS-labelled dataset using only one feature template $\phi^{(0)}_{l,\tau} = p(l|\mathcal{I}, \tau)$, with associated thresholds θ randomly generated from a uniform distribution in the $[0; 1]$ interval.

As illustrated on Figure 3, we also study the importance of using multiple available expression datasets for learning the first layer of trees (*i.e.* LEP representation). We can either train the models on a specific categorical expression database, or merge the datasets to learn LEP representation from all the available corpus ($M1$). Finally, we can also learn LEPs separately from the different categorical expression datasets and use a concatenation of the LEP feature vectors as an input for the second (AU prediction) tree layer ($M2$). Section 5.4 shows that those two approaches enhance the predictive power of the AU detection framework. Furthermore, those two strategies can complement each other well. Indeed, $M1$ requires to simultaneously load multiple datasets at training time, $M2$ involves computing multiple LEP features for evaluation. Thus, a combination of those two strategies can be used to fulfil the target memory/time requirements.

Also note that we voluntarily keep the AU recognition layer simple so as to showcase the usefulness of LEP representation for the AU prediction task, as compared to low-level engineered descriptors and other state-of-the-art methods. However, as shown in other works on expression recognition [24], recent approaches such as multi-task formulations (e.g. training a single RF for predicting multiple AUs) can significantly improve performances.

4.4.2 Confidence assessment for AU prediction

Because AUs are defined locally, chances are that AU activation relatively to an occluded area can not be predicted at all. Thus, we use the weights outputted by the autoencoder network to automatically derive a confidence score relatively to each AU indexed by m. To this end, we define as $N^{(m)}_{l,\tau}$ the number of times that the LEP feature $\phi^{(0)}_{l,\tau}$ was selected for splitting at the root of the trees, among all trees in the forest. This is highlighted on Figure 4. The reason for exclusively considering features at the root of the trees is that those features are selected from large numbers of training examples, as opposed to features from nodes deeper in the trees, that are essentially more noisy.

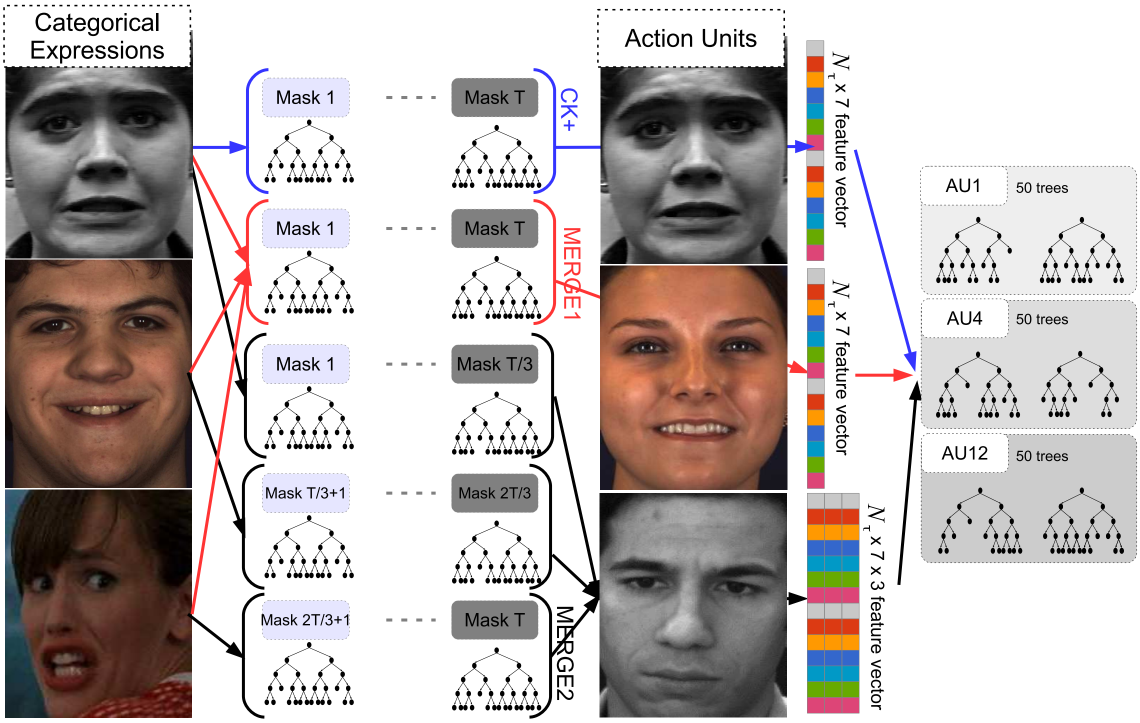

Figure 3: AU recognition using LEP features.

Note that, while most approaches focus on describing expressions as a combination of AUs, we can decompose each AU as a set of local expression predictions. For example, for AU1 (inner brow raiser) and AU2 (outer brow raiser), the most relevant LEPs are triangles corresponding to the inner and outer brows, associated with expression *surprise*, respectively. AU4 (brow lowerer) mainly uses triangles between the eyes associated with expression *anger*. AU9 (nose wrinkler) mainly uses triangles from the nose and cheek regions, associated with *disgust*. AU12 (lip corner puller) and AU20 (lip stretcher) respectively use triangles corresponding to lip corners with expressions *happiness* and *fear*.

We then define the AU-specific confidence measurement α_m for AU m as the sum of confidences $\alpha^{(\tau)}$ of triangles τ of the facial mesh, weighted by the proportion of LEP features from that triangle, that are used to describe the activation of AU m:

$$\alpha_m = \frac{\sum_\tau \alpha^{(\tau)} N_{l,\tau}^{(m)}}{\sum_\tau N_{l,\tau}^{(m)}} \tag{13}$$

Thus, the AU-specific confidence measurement is proportional to the confidence of the face regions that are the most useful for describing the activation of a specific AU. We show in the following section that such simple setting allows to highlight the cases were the AU predictions are deemed unreliable.

5 Experiments

In this section, we evaluate our approach on several FER benchmarks. Section 5.1 introduces the databases that are used for test. Section 5.2 sums up our experimental protocols and describe hyperparameter settings to ensure reproducibility of the results. In Section 5.3.1, we show results for FER on non-occluded data on three publicly available FER benchmarks that exhibit various degrees of difficulty, showing that our approach improves the state-of-the-art. Then, in Section 5.3.2, we report results on synthetically occluded images. Thus, we can precisely measure the robustness of our approach to occlusions. In Section 5.4 we give results of AU detection, showing that LEPs yield high predictive power for the task of AU detection compared

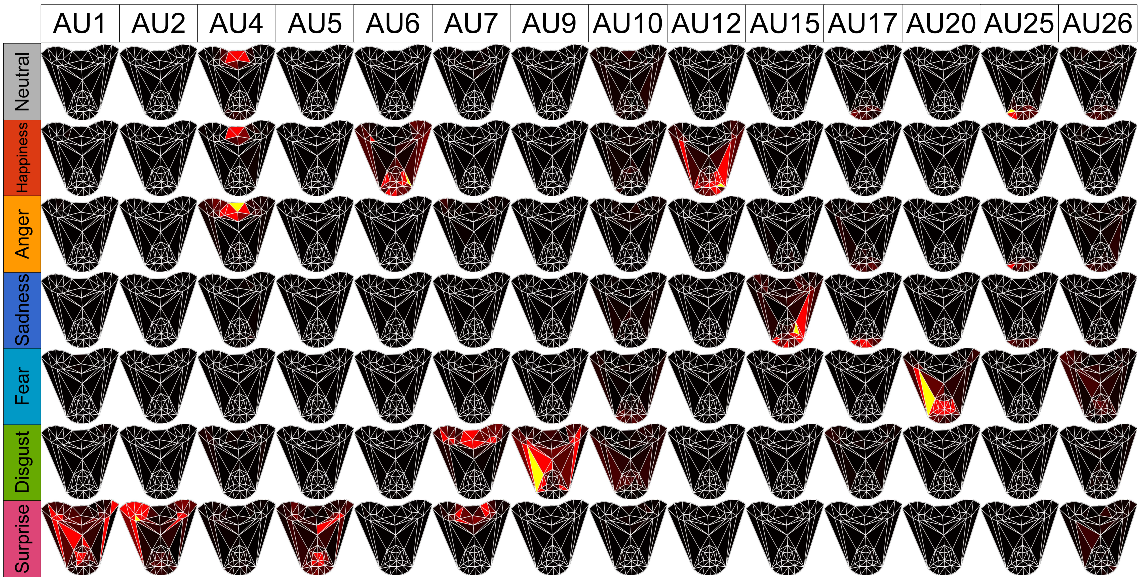

Figure 4: LEPs heat map for each of the 14 AUs (CK+ database). Only top-level LEP features are displayed for each AU. Best viewed in color.

to low-level features or state-of-the-art approaches. We also evaluate the relevance of the AU-specific confidence measurements. Lastly, Section 5.5 reports evidence of the real-time capacities of the proposed framework.

5.1 Datasets

5.1.1 Categorical expressions datasets

The CK+ or **Extended Cohn-Kanade database [2]** contains 123 subjects, each one displaying some of the 6 universal expressions (*anger*, *happiness*, *sadness*, *fear*, *digust* and *surprise*) plus the non-basic expression *contempt*. Expressions are prototypical and performed in a controlled lab environment with no head pose variation. As it is done in other approaches, we use the first (*neutral*) and three apex frames for each of the 327 sequences for 8-class categorical FER. As some approaches discard the frames labelled as *contempt*, we also report 7-class accuracy from 309 sequences.

The BU-4D or **BU-4DFE database [3]** contains 101 subjects, each one displaying 6 acted categorical facial expressions with moderate head pose variations. Expressions are still prototypical but they are performed with lower intensity and greater variability than in CK+, hence the lower baseline accuracy. Sequence duration is about 100 frames. As the database does not contain frame-wise expression, we manually select neutral and apex frames for each sequence.

The SFEW or **Static Facial Expression in the Wild database [11]** contains 700 images from 95 subjects displaying 7 facial expressions in a real-world environment. Data was gathered from video clips using a semi-automatic labelling process. The strictly person-independent evaluation (SPI) benchmark is composed of two folds of (roughly) same size. As done in other approaches, we report cross-validation results averaged over the two folds.

5.1.2 Action Unit datasets

The CK+ database is also FACS-annotated, therefore we report results for the recognition of 14 of the most common AUs (AU1,2,4,5,6,7,9,10,12,15,17,20,25,26).

The BP4D database [4] contains 41 subjects. Each subject was asked to perform 8 tasks, each one supposed to give rise to 8 spontaneous expressions (*anger*, *happiness*, *sadness*, *fear*, *digust*, *surprise*, *embarrassment* or *pain*). In [4] the authors extracted subsequences of about 20 seconds for manual FACS annotations, arguing that these subsets contain the most expressive behaviors. As done in the litterature [4] we report results for recognition of 12 AUs (1,2,4,6,7,10, 12,14,15,17,23,24). We randomly extract 10000 images for training and evaluate the AU classifiers on the whole dataset.

The DISFA or **Denver Intensity of Spontaneous Facial Actions [35]** contains videos of 27 subjects with different ethnicities and genders that were recorded watching a 4-minute emotive video stimulus. Data have been manually labeled frame by frame for 12 AUs (1,2,4,5,6,9,12,15,17,20, 25,26) on a 6-level scale by a human expert, and verified by a second FACS coder. For the purpose of predicting AU occurrence, we consider AU which intensity is below 1 as non-activated. We randomly extract 6292 images for training and test on the 125832 images.

5.2 Experimental setup

5.2.1 Evaluation metrics

For both occluded and non-occluded scenarios of categorical FER we use the overall accuracy as a performance metric. We also report confusion matrices to show the discrepancies between recognition of the expression classes. For AU detection we use the area under the ROC curve (AUC) as a performance metric, as it is widely used in the literature because it is independent of the setting of a decision threshold. For all the experiments, RF classifiers are evaluated with Out-Of-Bag (OOB) error estimate, with bootstraps generated at the subject level to ensure that, for each tree, subjects used for training are not used for testing this specific tree. The OOB error, according to [32], is an unbiased estimate of the true generalization error. Moreover, as stated in [36] this estimate is generally more pessimistic than traditional (e.g. k-fold or leave-one-subject-out) cross-validation estimates, further reflecting the quality of the results. For AU recognition, LEPs are generated for Out-Of-Bag examples for each tree and AUs are evaluated with OOB error.

5.2.2 Hyperparameter setting

In order to decrease the variance of the error we train large collections of trees ($T_1 = 1000$ for LEP generation, $T_2 = 50$ for AU detection). For training the local models, we set the locality parameter R to 0.1 (which means that each local model uses $1/10$ of the face total surface) which provided good robustness to occlusions. Finally, we use 40 $\phi^{(1)}$, 40 $\phi^{(2)}$ and 160 $\phi^{(3)}$ features for learning LEPs, as well as 25 threshold evaluations per features. For AU detection, we examine 100 $\phi^{(0)}$ features at each node, each associated with 25 threshold values. Note however that the values of these hyperparameters (except for R) had very little influence on the performances. This is due to the complexity of the RF framework, in which individually weak trees (e.g. that are grown by only examining a few features per node) are generally less correlated, still outputting decent predictions when combined altogether.

For the occluded scenarios on CK+ and BU4D, the autoencoder networks are trained in a cross-database fashion (*i.e.* training on CK+ and testing on BU4D and vice versa). Lastly, on SFEW database, we use the autoencoder network trained on CK+, as SFEW embraces multiple examples of occluded faces.

5.3 Categorical FER

5.3.1 FER on non-occluded images

In Tables 1, 2, 3 we report the average accuracy obtained by our local subspace Random Forest (LS-RF) and the confidence-weighted version (WLS-RF). We also compare with standard RF (RS-RF).

Generally speaking, classification results of LS-RF are a little better than those of the RS-RF. Indeed, forcing the trees to be local allows to capture more diverse information. RS-RF relies quite heavily on the mouth region, but other areas (e.g. around the eyes, eyebrows and nose regions) may also convey information that can be captured by local models. Figure 5 displays the proportion of top-level features over all triangles of the face area.

While more than 90% of the features extracted by RS-RF are concentrated around the mouth, the repartition for LS-RF is more homogeneous. Hence, LS-RF is less prone to a misalignment of the mouth feature points, or to occlusions of the mouth region. Furthermore, weighting the local predictions (WLS-RF) using the confidence score from the autoencoder network allows to enhance the results on BU4D and SFEW.

Table 1: CK+ database. †: CK database

CK+	Protocol	7em	8em
LBP [8]	10-fold	88.9†	-
CSPL [9]	10-fold	89.9†	-
iMORF [10]	10-fold	-	90.0
AUDN [13]	10-fold	93.7	92.0
RS-RF	OOB	92.6	91.5
LS-RF	OOB	94.1	**93.4**
WLS-RF	OOB	**94.3**	**93.4**

Table 2: BU4D database

BU4D	Protocol	% Acc
BoMW [37]	10-fold	63.8
Geometric [38]	10-fold	68.3
LBP-TOP [39]	10-fold	71.6
2D FFDs [40]	10-fold	73.4
RS-RF	OOB	73.1
LS-RF	OOB	74.3
WLS-RF	OOB	**75.0**

Table 3: SFEW database

SFEW	% Acc
PHOG-LPQ [11]	19.0
DS-GPLVM [12]	24.7
AUDN [13]	30.1
Semi-Supervised [14]	34.9
RS-RF	35.7
LS-RF	35.6
WLS-RF	**37.1**

Table 4: Confusion matrix (CK+-8em)

	ne	ha	an	sa	fe	di	co	su
ne	92.4	0.3	0.9	0.6	1.2	0.6	3.97	0
ha	0	100	0	0	0	0	0	0
an	4.4	0	91.1	0	0	2.3	2.3	0
sa	22.6	0	0	77.4	0	0	0	0
fe	1.3	4	0	0	90.7	0	0	4
di	3.4	0	0.6	0	0	96.1	0	0
co	11.1	0	0	3.7	0	0	85.2	0
su	1.6	0	0	0	0.4	0	1.2	96.8

Table 5: Confusion matrix (BU4D)

	ne	ha	an	sa	fe	di	su
ne	89.5	0	1.8	4.4	0.9	0.9	2.6
ha	2	89.9	0	0	5	2	1
an	10.1	0	70.7	7.1	2	9.1	1
sa	11	0	15	71	3	0	0
fe	9.8	17.6	2.9	5.9	38.3	11.8	13.7
di	3	4	6.9	1	7.9	73.3	4
su	0	1	0	1	6.2	0	91.8

Table 6: Confusion matrix (SFEW)

	ne	ha	an	sa	fe	di	su
ne	50.2	8.8	9.0	10.0	2.0	16.9	3.1
ha	10.6	67.5	6.2	6.9	2.6	3.5	2.6
an	25.4	16.1	31.3	10.1	3.7	0.9	12.5
sa	21.2	21.2	8.1	22.2	7.1	9.1	11.1
fe	14.2	16.2	13.0	5.0	23.1	7.1	21.3
di	31.3	23.7	10.4	7.1	3.7	15.6	8.2
su	15.4	11.0	12.1	3.3	7.7	6.6	44.0

The reason is that subjects from those datasets exhibit uncommon morphological traits, occlusion or lighting patterns. As such, more emphasis is put on reliable local patterns, resulting in a better overall accuracy. It also explains why the accuracy is equivalent for LS-RF and WLS-RF on CK+ database, where there is less variability. On the three databases, LS-RF and WLS-RF models provide better results compared to state-of-the-art approaches, even though some of these use complex FFD or spatio-temporal features (LBP-TOP), or use additional unlabelled data for regularization [14]. Note however that the evaluation protocols are different for some of these approaches. For example, authors in [12] use only the texture information and not the provided landmarks.

Tables 4, 5, 6 show the confusion matrices of WLS-RF on CK+, BU4D and SFEW respectively. Generally speaking, expressions *neutral*, *happy* and *surprise* are mostly correctly recognized, as they involve the most recognizable patterns (smile or eyebrow raise). *Anger* and *disgust* are also accurately recognized on CK+ and BU4D but not so much on SFEW. *Sadness* and *fear* seems to be the most subtle ones, particularly on BU4D and SFEW where those expressions can be misclassified as *surprise* or *happy*, respectively.

5.3.2 FER on occluded face images

In order to assess the robustness of our system to partial face occlusion, we measured the average accuracy outputted by RS-RF, LS-RF and WLS-RF on CK+ (8 expressions) and BU4D (7 expressions) databases with synthetic occlusions. More precisely, for each image we use the feature points tracked on non-occluded images to highlight the eyes and mouth regions. We then overlay a noisy pattern (see Figure 6), which is a more challenging setup than black boxes used in [18, 19]. We add margins of 20 pixels to the bounding boxes to make sure we cover the whole eyes (with eyebrows, as it represents the most valuable source of

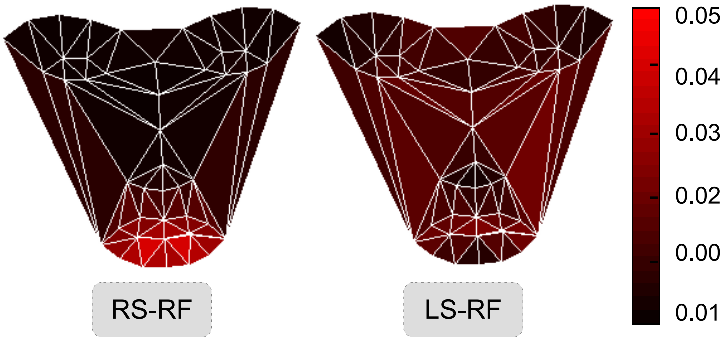

Figure 5: Proportion of top-level (tree root) features per triangle. Best viewed in color.

information from the eye region) and mouth region. Finally, we align the feature points on the occluded sequences.

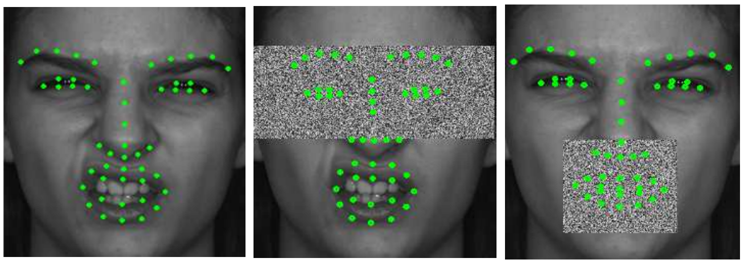

Figure 6: Examples of occluded faces from BU4D with aligned feature points. Left: non-occluded, middle: eyes occluded, right: mouth occluded. Also notice how the presence of an occlusion may have a critical effect on the quality of the feature point alignment.

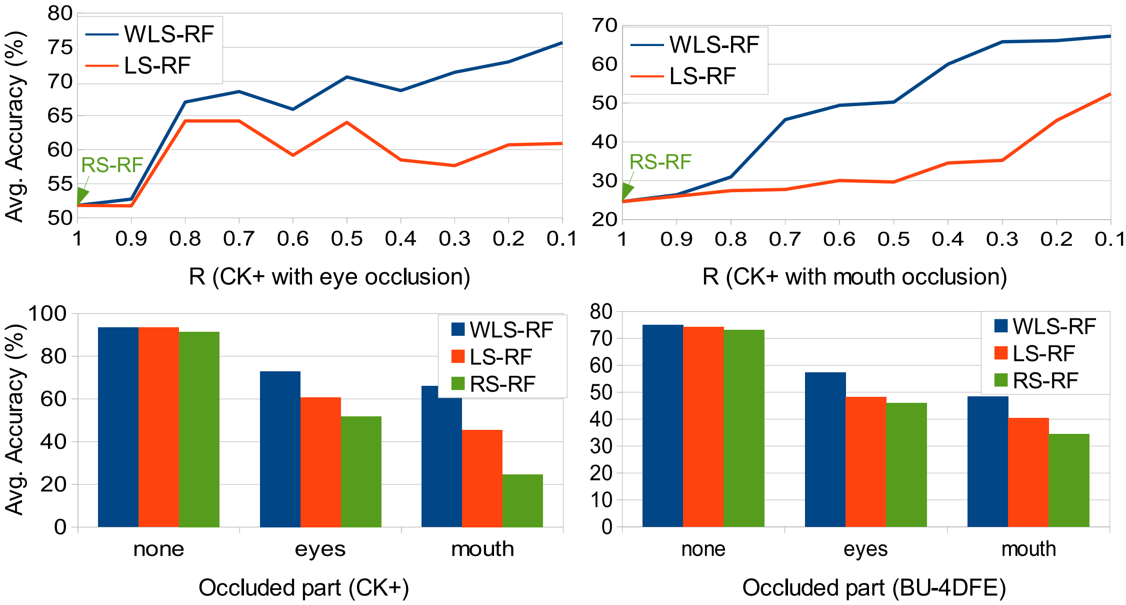

Figure 7: Accuracy outputted on occluded CK+ and BU4D databases

Influence of R: Graphs of Figure 7 show the variation of average accuracy *vs.* hyperparameter R that controls the locality of the trees, respectively under eyes and mouth occlusion on CK+ database. Performances of RS-RF fall heavily when the mouth is occluded (from 91.5% to 25.4%), as observed in [18]. This further proves that the global model relies essentially on mouth features to decipher facial expressions. Forcing the trees to be more local (e.g. setting R to 0.1 or 0.2) allows to capture more diverse cues from multiple facial areas, ensuring more robustness to mouth occlusion. It also explains why LS-RF models with $R = 0.8 - 0.5$ can already be quite robust to eyes occlusions, as the majority of the information used on such models likely comes from mouth area. Nevertheless, on those two occlusion scenarios, WLS-RF achieves a substantially better accuracy than the unweighted local models. Figure 7 also shows the accuracy comparison for both eyes and mouth occlusion scenarios on CK+ and BU4D, with $R = 0.1$. On the two databases, LS-RF is more robust to partial occlusions than RS-RF. Furthermore, WLS-RF also provides better accuracy than both LS-RF and RS-RF. Overall, the recognition percentage for WLS-RF under mouth occlusion is 67.1% against 30.3% for [18] in the case where classifiers are trained on non-occluded faces and tested on occluded ones.

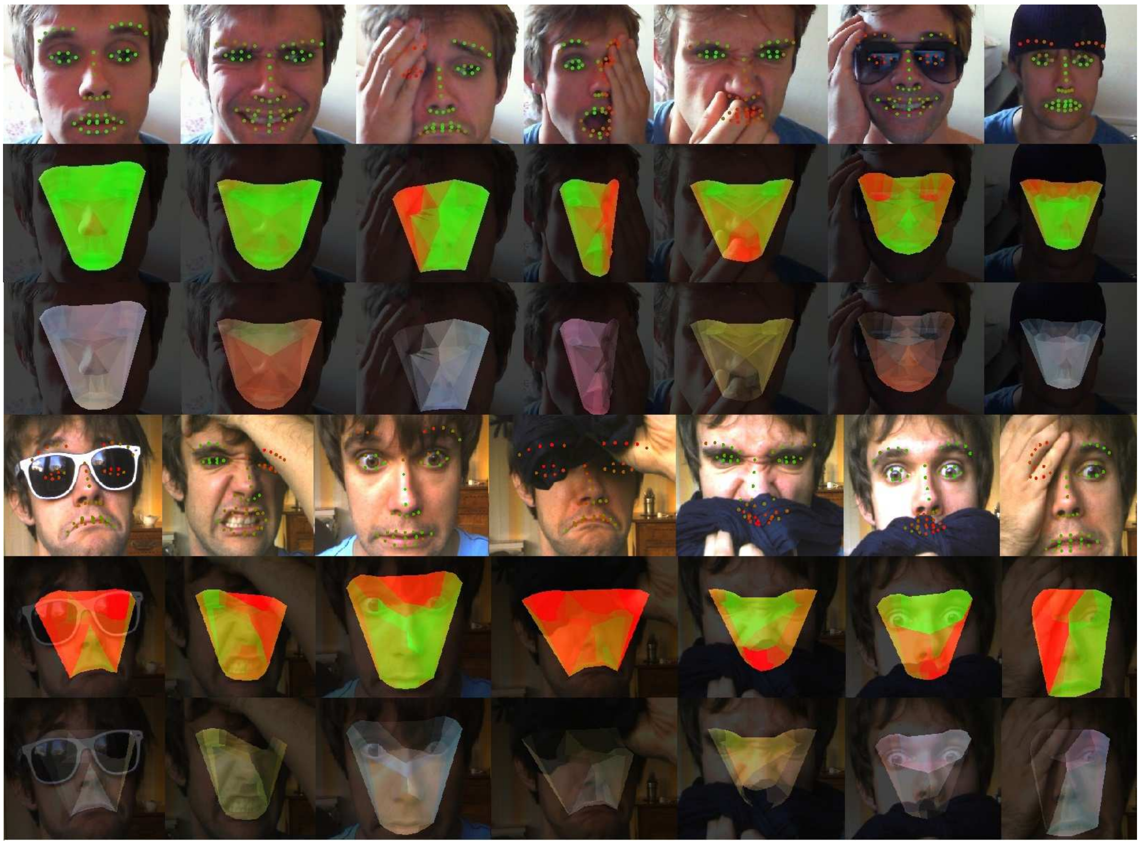

Figure 8: Examples of local FER under realistic occlusions. Top rows: point-wise confidence scores (red: low confidence, green: high confidence). Middle rows: triangle-wise scores. Bottom rows: weighted local classification (transparent for low confidences, gray for neutral, red for *happy*, yellow for *angry*, blue for *sad*, cyan for *fear*, green for *disgust* and magenta for *surprise*). Best viewed in color.

Realistic occlusions: our occlusion model is however quite "boring", in the sense that the occluding noisy patterns are not realistic. For that matter, and because there is currently no FER database that includes annotated partial occlusion ground truth, we also present on Figure 8 qualitative results on more realistic occlusions. Notice how the autoencoder network (learnt on CK+) assign high confidences (green) to non-occluded feature points, whereas examples that lie further from the captured manifold (e.g. because of lighting conditions, self-occlusion with a hand or with an accessory) are given lower values (red). The corresponding triangles are thus downweighted for FER and appear transparent on the last row. Also note that different facial regions can vote for different expressions, as shown on the second column (*happy*+*angry*/*disgust*).

5.4 AU detection

5.4.1 Merging multiple datasets

In this section we present results for AU detection using LEP features. Table 7 shows comparison of AUC for the prediction of AU activations on CK+ database obtained with LEPs trained on CK+, BU4D and SFEW databases, as well as models obtained *via* the $M1$ and $M2$ strategies.

For nearly every AU on CK+, the best AUC score is provided by the $M2$ strategy. However LEPs trained on CK+ only as well as the $M1$ strategy also provide good prediction results. LEPs trained solely on BU4D and SFEW seems a bit lackluster, but using the additional categorical expression data in addition to data from CK+ can be beneficial for prediction accuracy. Interestingly, on BP4D, LEPs trained on CK+ only seem to have a slight edge over the two LEPs models trained using all the available data. However, the $M2$ strategy and, to a lesser extent, $M1$ and training on BU4D only, provide close performances. Furthermore,

Table 7: AUC scores on CK+, BP4D and DISFA databases

	CK+					BP4D					DISFA				
AU	M1	M2	CK+	BU4D	SFEW	M1	M2	CK+	BU4D	SFEW	M1	M2	CK+	BU4D	SFEW
AU1	97.9	**98.4**	98.4	94.7	93.3	59.6	62.7	**63.6**	60.9	52.0	66.1	68.4	**71.3**	57.6	66.6
AU2	98	**98.2**	97.7	97.5	97.2	65.4	64.8	62.3	**66.0**	53.0	53.8	55.2	**67.3**	59.3	59.4
AU4	93.3	**95.4**	94.8	83.1	85.6	**68.7**	63.8	64.4	64.4	55.3	66.7	66.7	**67.3**	64.0	67.6
AU5	94	**97.5**	95.5	93.2	95	-	-	-	-	-	84.2	85.6	73.3	**88.6**	73.7
AU6	95.4	**95.7**	95.5	94.3	94.9	**83.1**	81.8	82.6	78.5	77.1	89.1	86.0	**89.2**	86.8	85.1
AU7	89.1	**90.2**	89.6	88.1	83	**76.8**	75.0	73.6	72.6	65.0	-	-	-	-	-
AU9	97.9	**99.3**	98.7	98.5	94.8	-	-	-	-	-	**79.0**	77.0	75.4	74.0	53.4
AU10	83.7	85.6	**86.5**	78.4	81.7	83.7	**83.8**	83.3	81.0	78.6	-	-	-	-	-
AU12	**97.6**	96	96.2	96	96.5	89.9	**90.0**	89.8	88.0	87.2	**95.5**	92.9	93.6	92.8	91.8
AU14	-	-	-	-	-	65.2	66.4	63.7	**66.5**	64.9	-	-	-	-	-
AU15	**91**	88.9	88.3	79	79.5	56.8	58.4	**58.5**	57.7	56.0	**69.5**	64.5	63.6	68.8	61.7
AU17	93.9	**95.1**	93.4	81.5	86.4	55.8	65.7	**68.9**	65.1	60.6	**67.8**	61.2	53.5	59.1	58.8
AU20	91.9	93.8	**94.5**	88.5	85.8	-	-	-	-	-	**65.0**	58.5	50.2	55.5	61.9
AU23	-	-	-	-	-	50.1	57.2	**60.2**	57.5	54.2	-	-	-	-	-
AU24	-	-	-	-	-	69.6	77.4	**78.2**	77.7	68.4	-	-	-	-	-
AU25	99	**99.1**	98.8	87.1	97.4	-	-	-	-	-	94.8	95.0	94.0	**95.6**	80.0
AU26	75.7	**81.2**	79.7	74.9	73.4	-	-	-	-	-	79.3	**81.4**	75.6	78.5	71.5
Avg	92.7	**93.7**	93.4	88.2	88.9	68.8	70.6	**70.8**	69.6	64.3	**75.9**	74.4	72.9	73.4	69.3

Table 8: Comparison with other works

Database	AUC(ours)	AUC(Other works)
CK+(14AU)	**93.7**	91.7 (SHTL [25])
BP4D(12AU)	**70.8**	68.9 (LBP-TOP [4])
DISFA(12AU)	**75.9**	75.7 (Multi-label CNN [41])

on the DISFA dataset, the $M1$ and the $M2$ LEPs models provide the highest AUC. Overall, the $M2$ and $M1$ models seem to perform better, followed by the models trained on CK+. This proves that AU detection can benefit from additional training data labelled with categorical expressions. Finally, LEPs trained on SFEW did not perform very well, probably due to the fact that the database embraces too much variability for too few training data. Thus, the categorical expressions can not be captured adequately, as can be seen from the low accuracies showed in Section 5.3.1.

5.4.2 Comparison with state-of-the-art approaches

Table 8 reports the overall best AUC obtained on the three datasets. It also draws a comparison between the AUC scores obtained using our method and results reported in recent publications involving similar protocols (same databases and sets of AUs, same intensity threshold for AU occurrence on DISFA). Our approach provides better results than SHTL [25] on CK+, as well as accuracy similar to the multi-label CNN introduced in [41] on DISFA. Furthermore, it provides better performance than baselines LBP-TOP features used in [4] on BP4D. This demonstrates that even though the AU detection pipeline is not meant to be optimal, LEPs learned on large amounts of categorical expression data yield high discriminative power for AU detection tasks. As such, an interesting direction would be to take into account the correlations between the tasks within a multi-output framework.

5.4.3 Relevance of AU confidence assessment

In order to assess the relevance of the AU-specific confidence measurement, we evaluated its average value on the occluded versions of the CK+ and BU4D databases generated in Section 5.3.2 for occlusion handling in categorical FER. From a general perspective, as can be seen on Figure 9, low confidence measurements can be observed for AUs from the upper face region on the two scenarios involving eye occlusion. The same holds for AUs from the lower face region and the "mouth occluded" scenario, whereas the confidence scores are significantly higher in the non-occluded case. Interestingly, confidence scores for AU6 (cheek raiser) and, to a lesser extent, AU9 (nose wrinkle), are quite low even in the "mouth occluded" case. Indeed, as can be witnessed on Figure 4.4, the confidence measurement for these AUs also use LEP features from the nose and mouth area.

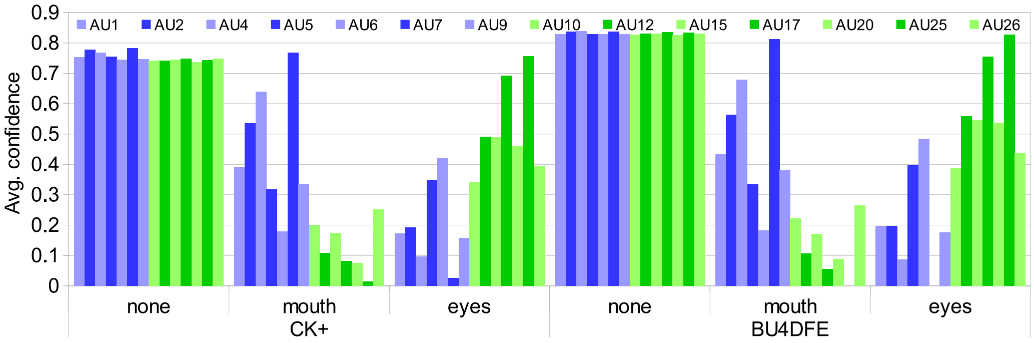

Figure 9: AU confidence scores outputted on occluded CK+ and BU4D database

Table 9: Measured evaluation time per processing step (in milliseconds)

Processing step	time (ms)
Feature point alignment	10
Integral channels computation	2
Confidence weights computation	11
LEP computation (1000 trees)	7
12 AU detection (50 trees)	1
Total	**31**

5.5 Computational load

The proposed framework for occlusion-robust FER (WLS-RF) and AU detection operates in real-time on video streams, even with large tree collections. Table 9 displays the elapsed time for each step of the evaluation pipeline. The test was performed on an Intel Core I7-4770 CPU on a single-thread C++/OpenCV implementation.

It appears that the feature point alignment and confidence weight generation steps are the bottleneck of the system in term of computational load. However the computational load for the former can be reduced by the use of more efficient alignment algorithms such as the one in [34]. As for the confidence weights, the computation time can be significantly reduced by a proper multithreading (e.g. computing the confidence for each feature point in parallel). As it is, the framework already runs at more than 30 fps even with large collections of trees. As for training, learning LEPs with 1000 trees on a big database (BU4D containing more than 8000 face images) took approximately three hours without parallelization. Training the hierarchical autoencoder network took half a day and learning the 12 AU detectors on DISFA database with 50 trees required one hour on the same I7-4770 CPU using a loose C++ implementation. Thus, our approach scales well both in terms of training and testing times, especially when compared to recent deep learning algorithms [41] for feature representation and learning.

6 Conclusion and perspectives

In this paper, we proposed a new high-level expression-driven LEP representation. LEPs are obtained from training Random Forests upon spatially defined local subspaces of the face. Extensive experiments on multiple datasets highlight the fact that the proposed representation improves the state-of-the-art for categorical FER and yields useful descriptive power for AU occurrence prediction. Furthermore, we introduced a hierarchical autoencoder network to model the manifold around specific facial feature points. We showed that the provided reconstruction error could effectively be used as a confidence measurement to weight the prediction outputted by the local trees. The proposed WLS-RF framework significantly adds robustness to partial face occlusions.

The ideas introduced in this work open a lot of interesting directions for future works on face analysis.

First, note that the confidence weights are representative of the spatially defined local manifold of the training data. Thus, these confidence values can be used to determine which parts of the face are the most reliable in a general way (e.g. to address unpredicted illumination patterns or head pose variations), and are not limited to occlusion handling. Furthermore, we could inject confidence weights into the feature point alignment framework [26] to enhance the robustness of the feature point alignment w.r.t. occlusions. Compared to a discriminative approach using synthetic data [17], our manifold learning approach could in theory more efficiently deal with realistic occlusions. Moreover, the applications of LEPs for AU detection and intensity estimation are multiple. First, it would be interesting to learn LEPs using more expression data such as the datasets introduced in [42, 43], possibly with a more complex integration strategy. Also, it could be interesting to investigate the impact of using a more fine-grained facial mesh for FER and AU detection or intensity estimation using LEP representation, as it was done in [44] for dense facial feature point alignment. Last but not least, the idea of learning Random Forests upon spatially defined local subspaces instead of random subspaces is not limited to face analysis, and could theoretically be applied in other fields such as gesture recognition.

References

[1] P Ekman and W Friesen. Constants across cultures in the face and emotion. *Journal of personality and social psychology*, 17(2):124, 1971.

[2] Patrick Lucey, Jeffrey F Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. The extended cohn-kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression. In *CVPR Workshops*, pages 94–101, 2010.

[3] Lijun Yin, Xiaochen Chen, Yi Sun, Tony Worm, and Michael Reale. A high-resolution 3D dynamic facial expression database. In *FG*, pages 1–6, 2008.

[4] Xing Zhang, Lijun Yin, Jeffrey F Cohn, Shaun Canavan, Michael Reale, Andy Horowitz, Peng Liu, and Jeffrey M Girard. BP4D-spontaneous: a high-resolution spontaneous 3D dynamic facial expression database. *IVC*, 32(10):692–706, 2014.

[5] Mark K Greenwald, Edwin W Cook, and Peter J Lang. Affective judgment and psychophysiological response: Dimensional covariation in the evaluation of pictorial stimuli. *Journal of psychophysiology*, 3(1):51–64, 1989.

[6] Paul Ekman and Wallace V Friesen. Facial action coding system. 1977.

[7] Zhihong Zeng, Maja Pantic, Glenn I Roisman, and Thomas S Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *PAMI*, 31(1):39–58, 2009.

[8] Caifeng Shan, Shaogang Gong, and Peter W McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *IVC*, 27(6):803–816, 2009.

[9] Lin Zhong, Qingshan Liu, Peng Yang, Bo Liu, Junzhou Huang, and Dimitris N Metaxas. Learning active facial patches for expression analysis. In *CVPR*, pages 2562–2569, 2012.

[10] Xiaowei Zhao, Tae-Kyun Kim, and Wenhan Luo. Unified face analysis by iterative multi-output random forests. In *CVPR*, pages 1765–1772, 2014.

[11] Abhinav Dhall, Roland Goecke, Simon Lucey, and Tom Gedeon. Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In *ICCV Workshops*, pages 2106–2112, 2011.

[12] Stefanos Eleftheriadis, Ognjen Rudovic, and Maja Pantic. Discriminative shared gaussian processes for multiview and view-invariant facial expression recognition. *TIP*, 24(1):189–204, 2015.

[13] Mengyi Liu, Shaoxin Li, Shiguang Shan, and Xilin Chen. Au-inspired deep networks for facial expression feature learning. *Neurocomputing*, 159:126–136, 2015.

[14] Mengyi Liu, Shaoxin Li, Shiguang Shan, and Xilin Chen. Enhancing expression recognition in the wild with unlabeled reference data. In *ACCV*, pages 577–588. 2013.

[15] Irene Kotsia, Ioan Buciu, and Ioannis Pitas. An analysis of facial expression recognition under partial facial image occlusion. *IVC*, 26(7):1052–1067, 2008.

[16] Shane F Cotter. Sparse representation for accurate classification of corrupted and occluded facial expressions. In *ICASSP*, pages 838–841, 2010.

[17] Golnaz Ghiasi and Charless C Fowlkes. Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model. In *CVPR*, pages 1899–1906, 2014.

[18] Ligang Zhang, Dian Tjondronegoro, and Vinod Chandran. Random Gabor based templates for facial expression recognition in images with facial occlusion. *Neurocomputing*, 145:451–464, 2014.

[19] Xiaohua Huang, Guoying Zhao, Wenming Zheng, and Matti Pietikäinen. Towards a dynamic expression recognition system under facial occlusion. *Pattern Recognition Letters*, 33(16):2181–2191, 2012.

[20] Marc Aurelio Ranzato, Joshua Susskind, Volodymyr Mnih, and Geoffrey Hinton. On deep generative models with applications to recognition. In *CVPR*, pages 2857–2864, 2011.

[21] Bihan Jiang, Michel F Valstar, and Maja Pantic. Action unit detection using sparse appearance descriptors in space-time video volumes. In *FG*, pages 314–321, 2011.

[22] Thibaud Sénéchal, Vincent Rapp, Hanan Salam, Renaud Seguier, Kevin Bailly, and Lionel Prevost. Facial action recognition combining heterogeneous features via multikernel learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42(4):993–1005, 2012.

[23] Wen-Sheng Chu, Fernando De la Torre, and Jeffrey F Cohn. Selective transfer machine for personalized facial action unit detection. In *CVPR*, pages 3515–3522, 2013.

[24] Jeremie Nicolle, Kevin Bailly, and Mohamed Chetouani. Facial action unit intensity prediction via hard multi-task metric learning for kernel regression. In *FG*, 2015.

[25] Adria Ruiz, Joost Van de Weijer, and Xavier Binefa. From emotions to action units with hidden and semi-hidden-task learning. In *IEEE International Conference on Computer Vision*, 2015.

[26] Xuehan Xiong and Fernando De la Torre. Supervised descent method and its applications to face alignment. In *CVPR*, pages 532–539, 2013.

[27] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[28] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *JMLR*, 11:3371–3408, 2010.

[29] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *ICML*, pages 833–840, 2011.

[30] Yuru Pei, Tae-Kyun Kim, and Hongbin Zha. Unsupervised random forest manifold alignment for lipreading. In *ICCV*, pages 129–136, 2013.

[31] Piotr Dollár, Zhuowen Tu, Pietro Perona, and Serge Belongie. Integral channel features. In *BMVC*, 2009.

[32] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[33] Arnaud Dapogny, Kevin Bailly, and Severine Dubuisson. Pairwise conditional random forests for facial expression recognition. In *ICCV*, 2015.

[34] Shaoqing Ren, Xudong Cao, Yichen Wei, and Jian Sun. Face alignment at 3000 fps via regressing local binary features. In *CVPR*, pages 1685–1692, 2014.

[35] S Mohammad Mavadati, Mohammad H Mahoor, Kevin Bartlett, Philip Trinh, and Jeffrey F Cohn. DISFA: A spontaneous facial action intensity database. *IEEE Transactions on Affective Computing*, 4(2):151–160, 2013.

[36] Tom Bylander. Estimating generalization error on two-class datasets using out-of-bag estimates. *Machine Learning*, 48(1-3):287–297, 2002.

[37] Liefei Xu and Philippos Mordohai. Automatic facial expression recognition using bags of motion words. In *BMVC*, pages 1–13, 2010.

[38] Yi Sun and Lijun Yin. Facial expression recognition based on 3D dynamic range model sequences. In *ECCV*, pages 58–71. 2008.

[39] M Hayat, Mohammed Bennamoun, and A El-Sallam. Evaluation of spatiotemporal detectors and descriptors for facial expression recognition. In *International Conference on Human-System Interaction*, pages 43–47, 2012.

[40] Georgia Sandbach, Stefanos Zafeiriou, Maja Pantic, and Daniel Rueck. A dynamic approach to the recognition of 3D facial expressions and their temporal models. In *FG*, pages 406–413, 2011.

[41] Sayan Ghosh, Eugene Laksana, Stefan Scherer, and Louis-Philippe Morency. A multi-label convolutional neural network approach to cross-domain action unit detection. In *ACII*, 2015.

[42] Arman Savran, Neşe Alyüz, Hamdi Dibeklioğlu, Oya Çeliktutan, Berk Gökberk, Bülent Sankur, and Lale Akarun. Bosphorus database for 3d face analysis. In *Biometrics and Identity Management*, pages 47–56. 2008.

[43] F Wallhoff. Database with facial expressions and emotions from technical university of munich (feedtum), 2006.

[44] László A Jeni, Jeffrey F Cohn, and Takeo Kanade. Dense 3d face alignment from 2d videos in real-time, 2015.