2017 IEEE International Conference on Computer Vision

Recognition of Action Units in the Wild with Deep Nets and a New Global-Local Loss

C. Fabian Benitez-Quiroz Yan Wang Aleix M. Martinez
Dept. Electrical and Computer Engineering
The Ohio State University
{benitez-quiroz.1,wang.9021,martinez.158}@osu.edu

Abstract

Most previous algorithms for the recognition of Action Units (AUs) were trained on a small number of sample images. This was due to the limited amount of labeled data available at the time. This meant that data-hungry deep neural networks, which have shown their potential in other computer vision problems, could not be successfully trained to detect AUs. A recent publicly available database with close to a million labeled images has made this training possible. Image and individual variability (e.g., pose, scale, illumination, ethnicity) in this set is very large. Unfortunately, the labels in this dataset are not perfect (i.e., they are noisy), making convergence of deep nets difficult. To harness the richness of this dataset while being robust to the inaccuracies of the labels, we derive a novel global-local loss. This new loss function is shown to yield fast globally meaningful convergences and locally accurate results. Comparative results with those of the EmotioNet challenge demonstrate that our newly derived loss yields superior recognition of AUs than state-of-the-art algorithms.

1. Introduction

Deep neural networks are proven algorithms in object detection and classification [1, 2, 3, 4, 5]. However, this advantage is only evident when a large number of annotated images is available.

The present paper addresses the problem of learning to detect Action Units (AUs) in images of facial expressions of emotion "in the wild." AUs are the observable anatomical facial movements defining a facial expression [6]. Each observable anatomical facial movement is given a unique number. For example, AU 1 is used to define the upper movement of the inner section of the eyebrows, and AU 12 the pulling of the corners of the lips.

Manually annotating AUs in images is cumbersome and can only be done by trained professionals [7]. This has limited the number of manually labeled images available to researchers, with the largest datasets only including thousands of samples [8, 9, 10]. These labeled images do not provide enough image variability (illumination, pose, occlusions) and ethnicity to take advantage of the richness of possible functions (VC-dimensionality) represented in deep nets [11].

Recently an AU-annotated set of about a million images of facial expressions in the wild was made available [12]. A large number of images were annotated with AUs 1, 2, 4, 5, 6, 9, 12, 17, 25 and 26. These images include the necessary image and individual variability needed to harness the richness of functional representation of deep neural networks.

Unfortunately, the labels of the images in this dataset are given by an algorithm developed by the authors of the database. This yields inaccurate annotations – according to the authors of this database, the annotations are about 81% correct. These inaccuracies causes major convergence problems in deep nets.

In this paper, *we derive a new global-local (GL) loss that can circumvent this problem of inaccurate labels while yielding accurate recognition results*.

Previous loss functions used either a global or local approach [13, 14, 15]. Global approaches take advantage of the overall structure of the object, yielding fast model to image fitting and globally consistent results. Thus, global loss functions facilitate convergence, but yield less accurate results. Local loss functions yield significantly more accurate results, but require large numbers of very accurate labels to converge.

Our newly derived GL-loss combines the idea of a local loss function, to emphasize the importance of accurate detections/recognitions, with a global loss, to yield a fast, meaningful convergence.

We use this newly derived criterion on deep convolutional neural networks (CNNs). A schematic comparison of a typical CNN versus ours, which employs our derived GL-criterion (GL-CNN), is shown in Figure 1. The resulting algorithm can be trained in a few hours. After training,

2380-7504/17 $31.00 © 2017 IEEE
DOI 10.1109/ICCV.2017.428

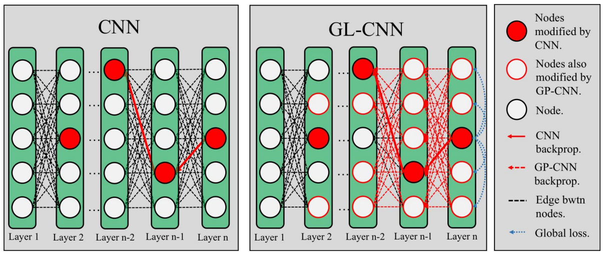

Figure 1. CNNs versus proposed GL-CNN. In multi-output CNNs, each output backpropagates its own local fit, i.e., backpropagation is based on how well this single output matches the expected (true) value. This is shown as filled red nodes and solid red lines on the left image. This process results in slow convergence rates and may lead to undesirable global fits, e.g., some of the outputs yield very good estimates, but others perform poorly. We derive a Global-Local (GL) loss function that optimizes each of the outputs of the network by backpropagating it locally *and* globally, shown as red circles and dashed lines in the neural network right of picture.

the system works in real time, i.e., >60 frames/s on an i7.

We compare our results to those of the EmotioNet challenge, which was completed earlier this year [16]. Our results show that the proposed GL-loss achieves meaningful convergences and results that are superior to those of state-of-the-art algorithms, including other deep nets.

Derivation of this proposed GL-criterion are in Section 2. The derived CNN and comparisons with other state-of-the-art nets are in Section 3. Comparative results are in Section 4. Conclusions are in Section 5.

2. Global-Local Loss

We present derivations of a global-local (GL) loss that can be efficiently used in deep nets for detection and recognition in images. We use this loss to train a deep CNN to recognize AUs. In our framework, a portion of the network is used to detect facial landmark points. These detections are concatenated with the output of the fully connected layer of the other components of the network to detect AUs.

2.1. Local fit

We define the image samples and their corresponding output variable as the set $\{(\mathbf{I}_1, \mathbf{y}_1), \dots, (\mathbf{I}_n, \mathbf{y}_n)\}$, where $\mathbf{I}_i \in \mathbb{R}^{l \times m}$ is a $l \times m$-pixel image of a face, $\mathbf{y}_i$ is the true (desirable) output, and n is the number of samples.

The output variable $\mathbf{y}_i$ can take many forms. For example, in the detection of 2D object landmark points in images, $\mathbf{y}_i$ is a vector of p 2D image coordinates $\mathbf{y}_i = (u_{i1}, v_{i1}, \dots, u_{ip}, v_{ip})^T$, $(u_{ij}, v_{ij})^T$ the j^{th} landmark points. In the recognition of AUs, the output variable corresponds to an indicator vector $\mathbf{y}_i = (y_{i1}, \dots, y_{iq})^T$, with $y_{ij} = 1$ if AU j is present in image $\mathbf{I}_i$ and $y_{ij} = -1$ when AU j is *not* present in that image. Figure 2 shows a face image with a set of 2D landmarks (yellow circles) and AU attributes (red boxes) as well as their corresponding output vectors.

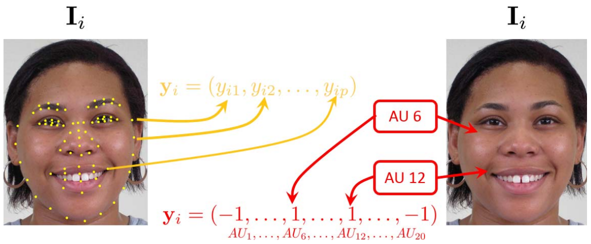
$\mathbf{I}_i$
$\mathbf{I}_i$
$\mathbf{y}_i = (y_{i1}, y_{i2}, \dots, y_{ip})$
AU 6
AU 12
$\mathbf{y}_i = (-1, \dots, 1, \dots, 1, \dots, -1)$
$AU_1, \dots, AU_6, \dots, AU_{12}, \dots, AU_{20}$

Figure 2. Detection of facial landmark points (left) and facial action units (AUs) (right). $\mathbf{I}_i$ is the i^{th} sample image, and $\mathbf{y}_i$ the desirable output. In landmark detection, the output corresponds to the 2D image coordinates of a set of p face landmark points. In AU recognition, the output vector indicates whether each AU is present (1) or not present (-1).

The goal of a computer vision system is to identify the vector of mapping functions $\mathbf{f}(\mathbf{I}_i, \mathbf{w}) = (f_1(\mathbf{I}_i, w_1), \dots, f_r(\mathbf{I}_i, w_r))^T$ that converts the input image $\mathbf{I}_i$ to an output vector $\mathbf{y}_i$ of detections or attributes, and $\mathbf{w} = (w_1, \dots, w_r)^T$ is the vector of parameters of these mapping functions. Note that $r = p$ and $\mathbf{f}(.) = (\hat{u}_{i1}, \hat{v}_{i1}, \dots, \hat{u}_{ip}, \hat{v}_{ip})^T$ in detection, with $f_j(\mathbf{I}_i, w_j) = (\hat{u}_{ij}, \hat{v}_{ij})^T$ the estimates of the 2D image coordinates u_{ij} and v_{ij}. Similarly, $r = q$ and $\mathbf{f}(.) = (\hat{y}_{i1}, \dots, \hat{y}_{iq})^T$ in the recognition of AUs, where $\hat{y}_{ij}$ is the estimate of whether AU j is present (1) or not (-1) in image $\mathbf{I}_i$, and q is the number of AUs.

For a fixed mapping function $\mathbf{f}(\mathbf{I}_i, \mathbf{w})$ (e.g., a CNN), the goal is to optimize $\mathbf{w}$; formally,

$$\mathcal{J}(\widetilde{\mathbf{w}}) = \min_{\mathbf{w}} \mathcal{L}_{\text{local}}(\mathbf{f}(\mathbf{I}_i, \mathbf{w}), \mathbf{y}_i), \tag{1}$$

where $\mathcal{L}_{\text{local}}(.)$ denotes the loss function. A classical solution for this loss function is the L^2-loss, defined as,

$$\mathcal{L}_{\text{local}}(\mathbf{f}(\mathbf{I}_i, \mathbf{w}), \mathbf{y}_i) = r^{-1} \sum_{j=1}^{r} \left(f_j(\mathbf{I}_i, w_j) - \mathbf{y}_{ij}\right)^2, \tag{2}$$

where $\mathbf{y}_{ij}$ is the j^{th} element of $\mathbf{y}_i$, which is $\mathbf{y}_{ij} \in \mathbb{R}^2$ in the detection of face landmark points and $y_{ij} \in \{-1, +1\}$ in the recognition of AUs.

Without loss of generality, we use $\mathbf{f}_i$ in lieu of $\mathbf{f}(\mathbf{I}_i, \mathbf{w})$ and f_{ij} instead of $f_j(\mathbf{I}_i, w_j)$. Note that the functions f_{ij} are the same for all i, but may be different for distinct values of j.

The above derivations correspond to a *local* fit. That is, (1) and (2) attempt to optimize the fit of each one of the outputs *independently* and then take the average fit over all outputs.

The above derived approach has several solutions, even for a fixed fitting error $\mathcal{J}(.)$. For example, the error can be equally distributed across *all* outputs $\|f_{ij} - \mathbf{y}_{ij}\|_2 \approx \|f_{ik} - \mathbf{y}_{ik}\|_2, \forall j, k$, where $\|.\|_2$ is the 2-norm of a vector. Or, most of the error may be in one (or a few) of the estimates: $\|f_{ij} - \mathbf{y}_{ij}\|_2 >> \|f_{ik} - \mathbf{y}_{ik}\|_2$ and $\|f_{ik} - \mathbf{y}_{ik}\|_2 \approx 0, \forall k \neq j$. In general, for a fixed fitting error, the latter example is less preferable, because it leads to large errors in one of the output variables. When this happens we say that the algorithm *did not converge* to a desirable result.

One solution to this problem is to add an additional constraint to minimize

$$\frac{2}{r(r+1)} \sum_{1 \leq j < k \leq r} |(f_{ij} - \mathbf{y}_{ij}) - (f_{ik} - \mathbf{y}_{ik})|^a \qquad (3)$$

with $a \geq 1$. However, this typically results in very slow training, limiting the amount of training data that can be efficiently used. By reducing the number of training samples, we generalize worse and typically obtain less accurate detections/recognitions [17, 18]. This is of course incompatible with our goal of using a million sample images.

Another typical problem of this equation is that it sometime leads to non-convergence (or convergence with very large fitting error $\mathcal{J}(.)$), because the constraint is not flexible enough for current optimization algorithms. This, in effect, reduces the VC-dimensionality of the net [19].

We solve the above defined problems in the next section by adding a *global* criterion that instead of slowing or halting convergence, it facilitates it.

2.2. Adding global structure

We define a set of constraints to add global structure to (1) by extending (2) to global descriptors.

It is key to note that the constraint in (2) is local because it measures the fit of each element of $\mathbf{y}_i$ (i.e., $\mathbf{y}_{ij}$) independently. By *local*, we mean that we only care about that specific result, Figure 1.

The same criterion can nonetheless be used to measure the fit of pairs of points; formally,

$$\mathcal{L}_{\text{pairs}}(\mathbf{f}_i, \mathbf{y}_i) = \frac{2}{r(r+1)} \sum_{1 \leq j < k \leq r} \left(g\left(h(f_{ij}), h(f_{ik})\right) - g\left(\mathbf{y}_{ij}, \mathbf{y}_{ik}\right)\right)^2, \qquad (4)$$

where $g(\mathbf{x}, \mathbf{z})$ is a function that computes the similarity between its two entries, and $h(.)$ scales the (unconstrained) output of the network into the appropriate value range.

In landmark detection, $h(f_{ij}) = f_{ij} \in \mathbb{R}^2$ and

$$g(\mathbf{x}, \mathbf{z}) = \|\mathbf{x} - \mathbf{z}\|_b \qquad (5)$$

is the b-norm of $\mathbf{x} - \mathbf{z}$ (e.g., the 2-norm, $g(\mathbf{x}, \mathbf{z}) = \sqrt{(\mathbf{x} - \mathbf{z})^T (\mathbf{x} - \mathbf{z})}$), where $\mathbf{x}$ and $\mathbf{z}$ are 2D vectors defining the image coordinates of two landmarks, Figure 2.

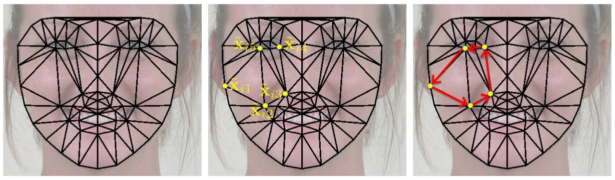
$\tilde{\mathbf{x}}_{i5}$
$\tilde{\mathbf{x}}_{i4}$
$\tilde{\mathbf{x}}_{i1}$
$\tilde{\mathbf{x}}_{i3}$
$\tilde{\mathbf{x}}_{i2}$

Figure 3. Facial landmark points define the Delaunary triangulation shown in the left image. Our proposed algorithm selects all possible sets of t landmarks, $\tilde{\mathbf{x}}_i = \{\tilde{\mathbf{x}}_{i1}, \dots, \tilde{\mathbf{x}}_{it}\}$, with $t = 1, \dots, t_{\text{max}}$. The middle image shows an example, with $t = 5$. Ordering the points counterclockwise allows us to compute the area of this polygon envelope (hull) with (9).

In AU recognition, $h(f_{ij}) = sign(f_{ij}) \in \{-1, +1\}$ and

$$g(x_{ij}, x_{ik}) = \begin{cases} 1, & \text{if } x_{ij} = x_{ik} \\ 0, & \text{otherwise,} \end{cases} \qquad (6)$$

where $sign(.)$ returns -1 if the input number is negative and $+1$ if this number is positive or zero. Recall, x_{ij} is 1 if AU j is present in image $\mathbf{I}_i$ and -1 if it is not present in that image. Hence, the function $h(.): \mathbb{R} \to \{-1, +1\}$.

Key to this process is to realize that (4) is no longer local, since it takes into account the *global* structure of each pair of elements, i.e., each pair of landmark points in detection and each pair of AUs in recognition, Figure 1. That is, in detection, we wish to use the information of the distance between all landmark points and, in recognition, we want to determine where pairs of AUs co-occur (meaning that the two are simultaneously present or not present in the sample image).

This global criterion can be easily extended to triplets. Formally,

$$\mathcal{L}_{\text{trip}}(\mathbf{f}_i, \mathbf{y}_i) = \binom{r}{3}^{-1} \sum_{1 \leq j < k < s \leq r} \left[g\left(h(f_{ij}), h(f_{ik}), h(f_{is})\right) - g\left(\mathbf{y}_{ij}, \mathbf{y}_{ik}, \mathbf{y}_{is}\right)\right]^2, \qquad (7)$$

where $g(\mathbf{x}, \mathbf{z}, \mathbf{u})$ is now a function that computes the similarity between its three entries.

In detection, this means we can compute the norm as in (5), e.g., $g(\mathbf{x}, \mathbf{z}, \mathbf{u}) = \|(\mathbf{x} - \mathbf{z}) + (\mathbf{z} - \mathbf{u})\|_b$, but we can also calculate the area of the triangle defined by each triplet; formally,

$$g(\mathbf{x}, \mathbf{z}, \mathbf{u}) = \frac{1}{2} \left|(\mathbf{x} - \mathbf{z}) \times (\mathbf{x} - \mathbf{u})\right|, \qquad (8)$$

where we assume the three landmark points are non-colinear.

These equations can be readily extended to four or more points. For instance, (8) can be extended to convex quadrilaterals as $g(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{v}) = \frac{1}{2} |(\mathbf{x} - \mathbf{u}) \times (\mathbf{z} - \mathbf{v})|$.

In the most general case, for t landmark points, we compute the area of the polygon envelope, i.e., a non-selfintersecting polygon contained by the t landmark points $\{\mathbf{x}_{i1}, \dots, \mathbf{x}_{it}\}$.[1] This polygon is given as follows.

First, the Delaunay triangulation of the facial landmark points is computed, Figure 3. The polygon envelop is easily obtained by connecting the lines of the set of t landmark points in counter-clockwise order, Figure 3. We denote this ordered set of landmark points $\tilde{\mathbf{x}}_i = \{\tilde{\mathbf{x}}_{i1}, \dots, \tilde{\mathbf{x}}_{it}\}$.

The area in $\tilde{\mathbf{x}}_i$ is given by,

$$g_a(\tilde{\mathbf{x}}_i) = \frac{1}{2}\left[\left(\sum_{k=1}^{t-1}\left(\tilde{x}_{ik1}\tilde{x}_{i(k+1)2} - \tilde{x}_{ik2}\tilde{x}_{i(k+1)1}\right)\right) + \right.$$
$$\left.(\tilde{x}_{it1}\tilde{x}_{i12} - \tilde{x}_{i12}\tilde{x}_{it1})\right], \tag{9}$$

where we used the subscript a in $g_a(.)$ to denote "area," and $\tilde{\mathbf{x}}_{ik} = (\tilde{x}_{ik1}, \tilde{x}_{ik2})^T$, Figure 3.

The result in (9) is easily obtained using Green's theorem as we show in the Supplementary Material. And, $\tilde{\mathbf{x}}_i$ can either be the t outputs of our CNN $\tilde{\mathbf{f}}_i = \{\tilde{f}_{ij}, \dots, \tilde{f}_{it}\}$ or the true values $\tilde{\mathbf{y}}_i = \{\tilde{\mathbf{y}}_{ij}, \dots, \tilde{\mathbf{y}}_{it}\}$.

We can also compute the global b-norm, $g_n(.)$, for the general case of t landmark points as,

$$g_n(\tilde{\mathbf{x}}_i) = \sum_{k=1}^{t-1} \|\tilde{x}_{ik1} - \tilde{x}_{i(k+1)2}\|_b. \tag{10}$$

The above derivations define the extension of $g(.)$ in (4) to three and more points in detection problems. Let us now see how this applies to recognition of AUs.

We want to compute the co-occurrence of three or more AUs in image $\mathbf{I}_i$. Formally, let $\tilde{\mathbf{x}}_i = \{\tilde{x}_{i1}, \dots, \tilde{x}_{it}\}$ be a set of t AUs, with $\tilde{x}_{ij} \in \{-1, +1\}$, $j = 1, \dots, t$, and

$$g_{AU}(\tilde{\mathbf{x}}_i) = \begin{cases} 1, & \text{if } \tilde{x}_{i1} = \dots = \tilde{x}_{it} \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

2.3. GL-loss

The final local-global (GL) loss function is given by,

$$\mathcal{L}(\mathbf{f}_i, \mathbf{y}_i) = \alpha_0 \mathcal{L}_{\text{local}}(\mathbf{f}_i, \mathbf{y}_i) + \mathcal{L}_{\text{global}}(\mathbf{f}_i, \mathbf{y}_i), \tag{12}$$

where the *global* loss $\mathcal{L}_{\text{global}}$ is defined as

$$\mathcal{L}_{\text{global}}(\mathbf{f}_i, \mathbf{y}_i) =$$
$$\sum_{t=1}^{t_{\max}} \alpha_t \left[g\left(h(\tilde{f}_{ij}), \dots, h(\tilde{f}_{it})\right) - g\left(\tilde{\mathbf{y}}_{ij}, \dots, \tilde{\mathbf{y}}_{it}\right)\right], \tag{13}$$

$g(.)$ is either $g_a(.)$ or $g_n(.)$ or both in detection and $g_{AU}(.)$ in recognition, and α_i are normalizing constants learned using cross-validation on the training set,.

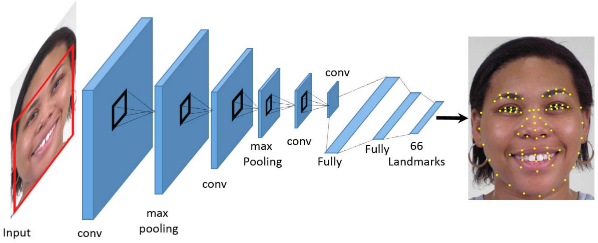

Figure 4. A schematic of the proposed algorithm as it applies to landmark detection. We use four convolutional layers, two max pooling layers and two fully-connected layers. The deep network for AU recognition is similarly defined (see text).

2.4. Backpropagation

To optimize the parameters of our CNN, $\mathbf{w}$, we need to compute

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \alpha_0 \frac{\partial \mathcal{L}_{\text{local}}}{\partial \mathbf{w}} + \frac{\partial \mathcal{L}_{\text{global}}}{\partial \mathbf{w}}. \tag{14}$$

The partial derivaties of the local loss is of course given by

$$\frac{\partial \mathcal{L}_{\text{local}}}{\partial w_j} = \frac{2}{r}\frac{\partial f_{ij}}{\partial w_j}\left(f_{ij} - \mathbf{y}_{ij}\right). \tag{15}$$

But how about the partial derivatives of the global loss?

In our definition of the global loss in (13) we used the mapping function $h(.)$. In landmark detection, $h(f_{ij}) = f_{ij}$ and, hence, the derivatives of the global loss have the same form as those of the local loss shown in (15). But for AU recognition, we used $h(f_{ij}) = sign(f_{ij}) \in \{-1, +1\}$. This function is not differentiable. Thankfully, we can solve this problem with the following simple redefinition $h(f_{ij}) = f_{ij}/\sqrt{f_{ij}^2 + \epsilon}$, for some small $\epsilon > 0$. Its partial derivative is $\partial h(f_{ij})/\partial w_j = 1/2 + 1/\sqrt{f_{ij} + \epsilon}$.

3. Proposed Deep Net

We define a deep convolutional neural network for the recognition of AUs. Our network consists of two parts: The first part of our network is used to detect a large number of facial landmark points, Figure 4. This was previously illustrated in Figure 2. These landmark points allow us to compute our GL-loss (12), as shown in Figure 3.

The results of these detections are normalized and concatenated with the output of the first fully connected layer of the second part of the network, Figure 5. This is to embed the location information of the landmarks into the network used to recognize AUs. This facilitates the detection of local shape changes typically observed in the expression of emotion [20]. This is done with (11) in the definition of the GL-loss.

Specifically, in our proposed network, nine layers are dedicated to the detection of facial landmark points (Figure 4), and the others are used to recognize AUs (Figure 5).

[1] Another solution is to compute the area of the convex hull of these t points. However, the structure of the convex hulls thus computed are very similar; polygons provide larger structural variety, facilitating higher accuracy in our results.

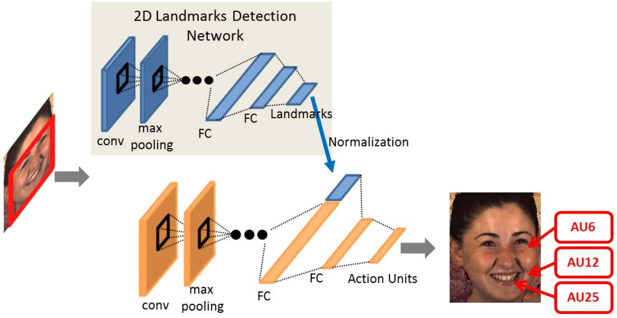

Figure 5. The proposed network for AU detection.

Layer type	Filter size	Number of filters
1st convoulutional	5×5	80
1st max pooling	2×2	–
2nd convoulutional	4×4	96
2nd max pooling	2×2	–
3rd convoulutional	3×3	128
4rd convoulutional	3×3	128
1st fully connected	1×1	1800
2nd fully connected	1×1	1000
Output	–	66×2

Table 1. Parameters used in the first part of the network to detect facial landmark points.

Let us first provide the details of the layers devoted to the detection of facial landmark points.

3.1. Facial landmark point detection

The general structure of our net is summarized in Figure 4. We use three convolutional layers, two max pooling layers and two fully connected layers. Following [4], we apply normalization, dropout, and rectified linear units (ReLU) at the end of each convolutional layer. Details on this first set of layers of our CNN are in Table 1.

The weights in these layers are optimized using backpropagation – using the derived GL-loss (12). The global loss is given by (9). The backpropagation equations are in Section 2.4.

We used this part of the network to detect a total of 66 facial landmark points.

As mentioned in the introduction of this paper, one advantage of the proposed GL-loss is that it can be efficiently trained on very large datasets. Since we wish to have a facial landmark detector invariant to any affine transformation and partial occlusions, we performed a data augmentation approach as follows.

Data augmentation: We generated additional images by applying two-dimensional affine transformations to the existing training set, i.e., scale, reflection, translation and rotation. Specifically, scale was taken between 2 and .5, rotation was $-10°$ to $10°$, and translation and reflection were randomly generated. To make the network more robust to partial occlusions, we added random occluding boxes of $d \times d$ pixels, with d between .2 and .4 times the inter-eye distance. 25% of our training images had partial occlusions.

3.2. AU recognition

As shown in Figure 5, the second part of the network combines the face appearance features with the landmark locations given by the first part of the network. Specifically, in the output of the first fully connected layer of the second part of the network, we concatenate the appearance image features with the normalized 66 automatically detected landmark points.

Formally, let $\mathbf{s}_i = (\mathbf{s}_{i1}^T, ..., \mathbf{s}_{ip}^T)^T$ be the vector of landmark points in the i^{th} sample image $(i = 1, ...n)$, where $\mathbf{s}_{ik} \in \mathbb{R}^2$ are the 2D image coordinates of the k^{th} landmark, and n is the number of sample images. Thus $\mathbf{s}_i \in \mathbb{R}^{132}$. All images are then normalized to have the same inter-eye distance of τ pixels. That is, $\hat{\mathbf{s}}_i = c\mathbf{s}_i$, where $c = \frac{\tau}{||\mathbf{l}-\mathbf{r}||_2}$, $\mathbf{l}$ and $\mathbf{r}$ are the image coordinates of the center of the left and right eye, $||\cdot||_2$ defines the $2-$norm of a vector, $\hat{\mathbf{s}}_i = (\hat{\mathbf{s}}_{i1}^T, ..., \hat{\mathbf{s}}_{ip}^T)^T$ and we use $\tau = 200$.

We further normalize the landmark points as $\hat{\mathbf{s}}'_{ik} = \mathbf{R}(\hat{\mathbf{s}}_{ik} - \hat{\mathbf{l}}) + \hat{\mathbf{l}}$, where $\hat{\mathbf{l}} = c\mathbf{l}$, and here we multiply the landmark points with a rotation matrix $\mathbf{R}$ to make the outer corner of left and right eyes match the horizontal line. Finally, we rescale and shift these values $\hat{\mathbf{s}}'_i$ to move the outer corner of left and right eyes to the pre-determined positions of $(.5, 0)$ and $(-.5, 0)$, respectively.

As for the structure of the deep net, we adopt that of GoogleNet [21]. Because the input of our network is the face image, we changed the size of the filter in the first layer to adapt to our input, and randomly initialize the weight for these filters. In order to embed landmark in our network, we also changed the number of filters in the first fully connected layer. And, we changed the number of filters for output as the number of AUs. Thus, a single deep net is employed to detect all AUs in images of facial expressions. The main modifications made to the architecture of GoogleNet are summarized in Table 2.

Of course, the loss used by our network is the GL-loss defined in the present paper.

The weights of this second part of the network are optimized using backpropagation on (12), with the global loss defined in (11). Details of these derivations are in Section 2.4.

We performed data augmentation by adding random noise to the 2D landmark points, and applying the same affine transformations described in Section 3.1.

3.3. Training the network

We used the labelled EmotioNet dataset of [12], which includes a large number of sample images with AUs 1, 2, 4,

Layer type	Filter size	Number of filters
1st convoulutional	3×3	64
1st fully connected	1×1	4096+132
Output	–	Active AUs

Table 2. Summary of the main changes made to the overall architecture of the GoogleNet of [21].

5, 6, 9, 12, 17, 25 and 26.

This database includes $950,000$ automatically labelled images, called the *training* set, and $25,000$ manually labeled images, dubbed the *verification* set. We use the training set to train the network until convergence is achieved. After that, we use the verification set to fine-tune the network. These sets were enlarged using the data augmentation approach described above.

Training was done in mini-batches. For this, the training images were divided into subsets of about $125,000$ each. Gradient descent was applied to each mini-batch. The verification set of $25,000$ manually annotated images was used as a final check after the first complete set of mini-batches finished. If needed, the data was divided into a new set of mini-batches and this process was repeated until the verification error was minimized. The derived GL-loss yielded very fast convergences in this framework. Adding the derived global term in the loss function reduced the number of epochs by more than 25% compared to when this term is not included.

3.4. Related work

AU recognition is an important task for emotion recognition. With the rapid development of deep neural network, some deep learning based methods were proposed in recent years. In [22], the authors proposed an attention map CNN, enhancing the regions of interest on the face when training the network. Instead of applying a regular CNN to the entire input image, Zhao *et al.* [23] proposed to add a region layer in CNN to identify specific regions for different AUs. However, limited annotated data makes the training of deep networks for AU recognition very challenging. More recently, a large annotated dataset with almost one million labeled images was published, which made this training possible. The top algorithm uses residual blocks and a sum of binary cross-entropy loss. Wang *et al.* [24] won the second place in the challenge and they use a multi-label softmax loss function. These algorithms suffer from convergence issues, as reported by the authors. Our proposed GL-loss yields fast convergences and better results, as shown in Section 4.

Neural nets generally use the square (L^2) loss [25]. The square-loss subtracts each output of the network to its corresponding desirable (true) values and squares it. The weights of the net responsible for this output are then modified using backpropagation. This local fitting process can easily lead to undesirable convergence due to the lack of global constraints, i.e., local criteria usually yield highly non-linear error functions, making it difficult to find a good local minimum. This is especially problematic in fine-grained detection and recognition problems (e.g., facial landmark detection and AU recognition) [26, 27]. Global constraints can be added to reduce the complexity of local fitting functions (e.g., with Glivenko-Cantelli estimates) [28], but this typically results in less accurate local fits than locally-defined loss functions.

A CNN solution is to define independent outputs for each detection/attribute using a region-based approach [29]. The same is true for the detection of landmark points [30] and the recognition of AUs [31]. Other constraints (e.g. spatio-temporal features where a video sequence is available) can be used to address the local-fitting problem too [32, 33].

The GL-loss function derived above diverges from these previous approaches in several and important ways. Our global constraints are based on the structure of the object (face). Specifically, in facial landmark detection, we defined the polygons of every set of t landmark points and derived an efficient algorithm to measure similarity between the output of the net and the true values. For AU recognition, we computed the co-occurrence of t AUs in each training image, i.e., either all t AUs are present or not. We showed that with the addition of these global constraints, a single global (*non-patch* based) CNN can be successfully defined. Our experimental results reported below demonstrate that this approach yields results superior to those of the state-of-the-art in the recognition of AUs.

4. Experimental Results

We provide extensive evaluations of the proposed approach and comparisons to state-of-the-art algorithms on the EmotioNet challenge [16]. The EmotioNet dataset [12] provides images of facial expressions "in the wild" with different natural illuminations, occlusions, poses and ethnicities, to name but a few characteristics.

Our evaluation is divided into four experiments. First, we present results following the evaluation protocol of the EmotioNet challenge. Second, we present results showing that the system is robust to scale changes. Third, we show that the proposed methodology is robust to occlusions. Finally, we show that the proposed algorithm is less sensitive to changes in pose than previous methods.

4.1. Facial expressions in the wild

We tested our algorithm following the protocol of the EmotioNet challenge [16]. The results reported in this section correspond to those obtained in the sequestered *testing* set, which was not available during training. A few qualitative examples of AU detections given by our algorithm can be found in Figure 6.

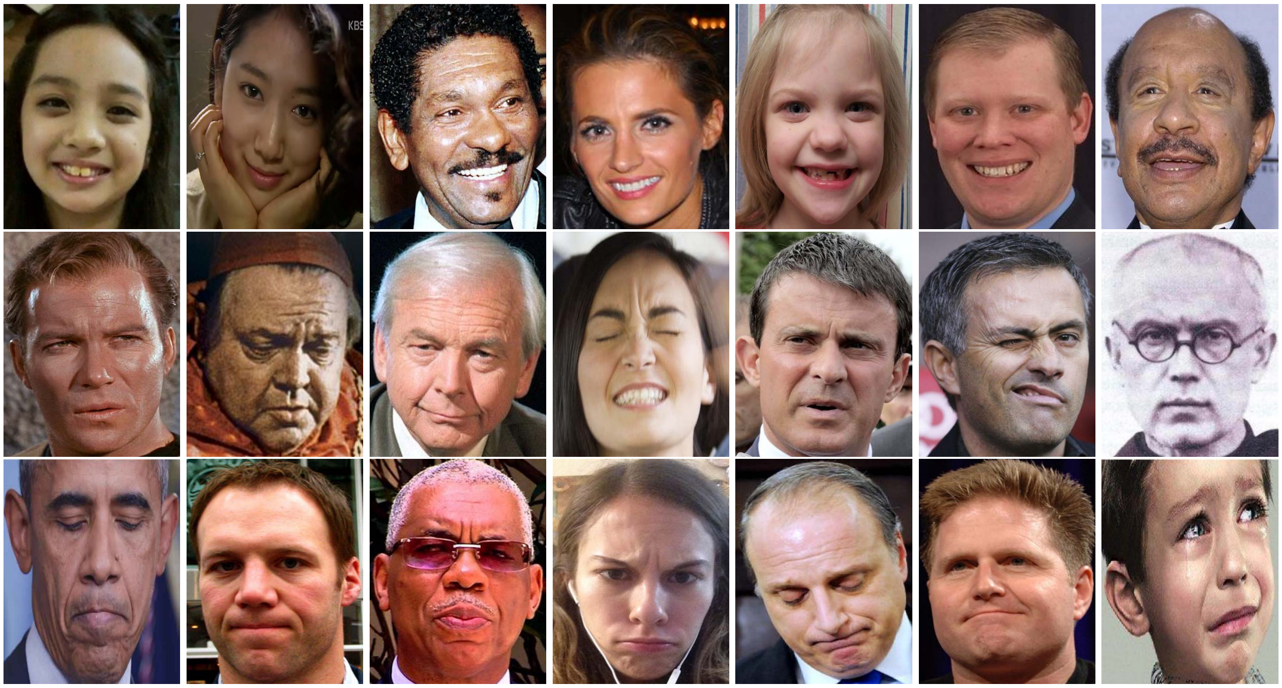

Figure 6. Examples of automatic annotations given by our deep net with our newly derived GL-loss. Top row shows images with AU 12 (lip corner puller), middle row shows images with AU 4 (brow lowerer), and lower row images with AU 17 (chin raiser).

The quantitative evaluation of the challenge includes two criteria. The first is F_1 score of AU i, defined as,

$$F_{1_i} = 2 \times \frac{Precision_i \times Recall_i}{Precision_i + Recall_i}, \tag{16}$$

where $Precision$ is the fraction of the automatic annotations of AU i that are correctly recognized (i.e., number of correct recognitions of AU i divided by the number of images with detected AU i), and $Recall$ is the number of correct recognitions of AU i over the actual number of images with AU i.

The second is accuracy of detection of AU i, defined as,

$$Accuracy_i = \frac{True\ positives_i + True\ negatives_i}{Total\ population}, \tag{17}$$

where $True\ positives_i$ are correctly identified test instances of AU i, $True\ negatives_i$ are test images correctly labeled as not belonging to AU i, and $Total\ population$ is the total number of test images.

In the EmotioNet challenge, these two criteria are averaged in a signal $Final\ score$, as

$$Final\ score_i = \frac{Accuracy_i + F_{1_i}}{2}. \tag{18}$$

Comparative results on the recognition of AUs are given in Figure 7. We provide comparative results as $Final\ score$ as well as F_1 score.

Figure 7 shows comparative results with: *a*) the deep neural network presented in this paper, *b*) the top 2 contenders in the EmotioNet Challenge [16] (both of them CNNs), and *d*) AlexNet using the standard *softmax loss*. Note that the implementation of AlexNet uses a different network to detect each AU i, whereas our proposed algorithm employs a single network. Training a single AlexNet to detect all AU (as in our approach) yielded much worse results than those reported in the figure.

4.2. Recognition at different scales and under occlusion

To evaluate the robustness of the system to scale, images were reduced to 1/2 and 1/4 of their original size. This yielded faces at different scales.

Figure 8 shows comparisons of the results of the proposed algorithm with the methods described in Section 4.1. As seen in the figure, image resolution did not affect the accuracy of the proposed algorithm.

Our experiments also included testing the robustness of the proposed algorithm to random occlusions in the test images. These correspond to black boxes of about $1/5^{th}$ the size of the face, as described in [16].

Figure 9 shows comparative results with the methods listed in Section 4.1. Sample AU detection in these images by our algorithm are in Figure 10.

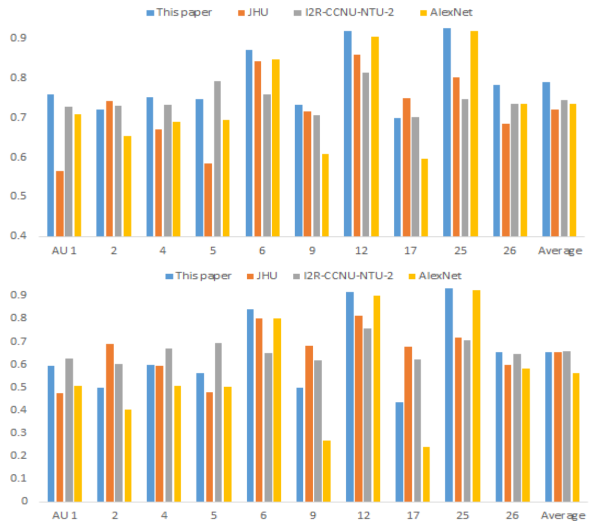

Figure 7. Results on the EmotionNet testing dataset. Top plot: Final scores given by (18). Bottom plot: F_1 scores calculated using (16).

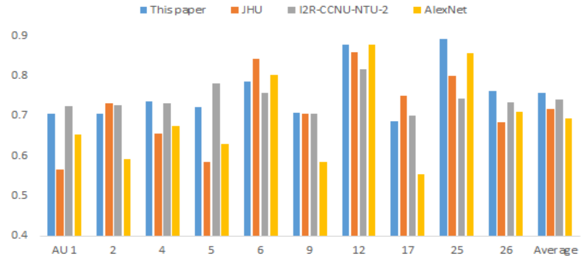

Figure 8. Average Final scores (given by (18)) for images at different scales.

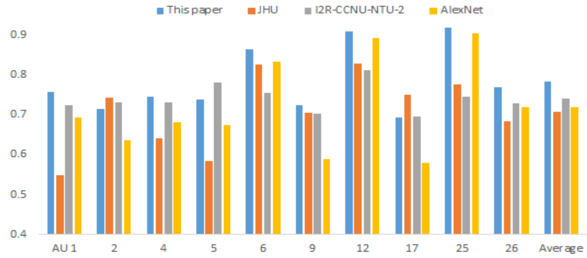

Figure 9. Final scores (equation (18)) for images with small occluders.

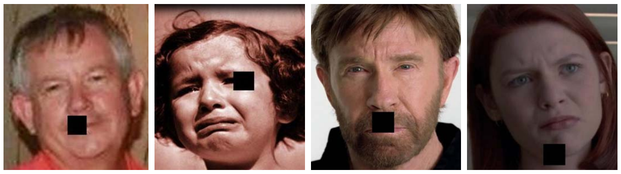

Figure 10. Examples of annotationsof AU 4 with images with artificial occlusions.

4.3. Pose invariance

Pose is known to be a major factor on the recognition of AUs in images of facial expressions "in the wild." We used the pose information of the test images provided by [16] to test the robustness of the derived algorithm in the recognition of AUs at different poses.

The results are in Figure 11. Pose is given in degrees and is defined as,

$$pose = \frac{|pitch| + |yaw|}{2}, \tag{19}$$

where $|.|$ is the absolute value.

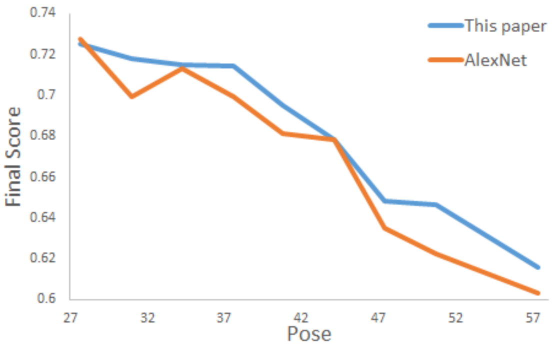

Figure 11. Average Final score as function of the Pose. Pose is defined in equation (19) and is given in degrees in the x-axis. The y-axis is the average Final score defined in (18).

5. Conclusions

We have derived a new Global-Local loss function for deep nets that can be efficiently used in detection of similar object landmark points of interest as well as recognition of object attributes. We have presented detailed derivations of the approach and several alternative models. The derived local+global loss yields accurate local results without the need to use patch-based approaches and results in fast and desirable convergences. Other than our theoretical arguments in favor of these claims, we have shown several experimental results demonstrating these abilities on our implementation of the derived algorithm in accuracy of detection/recognition and speed – our algorithm runs at > 60 frames/s. Our experimental results also demonstrate that the proposed GL-based algorithm outperforms other state-of-the-art methods in the recognition of action units. The derived loss achieves this by finding this structure, e.g., the co-articulation of AUs or the spatial arrangement of landmark points in a face.

Acknowledgements. Research supported by the National Institutes of Health, grant R01 DC 014498 and the Human Frontier Science Program, grant RGP0036/2016.

References

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, pp. 1–42, 2014. 1

[2] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*, pp. 740–755, Springer, 2014. 1

[3] X. Chen and A. Gupta, "Webly supervised learning of convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1431–1439, 2015. 1

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012. 1, 5

[5] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. Li, "Large-scale video classification with convolutional neural networks," in *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pp. 1725–1732, 2014. 1

[6] P. Ekman and W. V. Friesen, *Facial action coding system*. Consulting Psychologists Press, Stanford University, Palo Alto, 1977. 1

[7] P. Ekman and E. L. Rosenberg, *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS), 2nd Edition*. Oxford University Press, 2015. 1

[8] S. M. Mavadati, M. H. Mahoor, K. Bartlett, P. Trinh, and J. Cohn, "Disfa: A spontaneous facial action intensity database," *Affective Computing, IEEE Transactions on*, vol. 4, April 2013. 1

[9] J. F. Cohn and F. De la Torre, "Automated face analysis for affective computing," in *The Oxford Handbook of Affective Computing* (R. Calvo and S. D'Mello, eds.), p. 131, Oxford University Press, USA, 2014. 1

[10] D. McDuff, R. Kaliouby, T. Senechal, M. Amr, J. Cohn, and R. Picard, "Affectiva-mit facial expression dataset (am-fed): Naturalistic and spontaneous facial expressions collected," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 881–888, 2013. 1

[11] P. L. Bartlett, "The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network," *IEEE transactions on Information Theory*, vol. 44, no. 2, pp. 525–536, 1998. 1

[12] C. F. Benitez-Quiroz, R. Srinivasan, and A. M. Martinez, "Emotionet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 5, 6

[13] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997. 1

[14] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3476–3483, 2013. 1

[15] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang, "Semantic image segmentation via deep parsing network," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1377–1385, 2015. 1

[16] C. F. Benitez-Quiroz, R. Srinivasan, Q. Feng, Y. Wang, and A. M. Martinez, "Emotionet challenge: Recognition of facial expressions of emotion in the wild," *arXiv preprint arXiv:1703.01210*, 2017. 2, 6, 7, 8

[17] Y. Pang, S. Wang, and Y. Yuan, "Learning regularized lda by clustering," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 12, pp. 2191–2201, 2014. 3

[18] A. M. Martínez and A. C. Kak, "Pca versus lda," *IEEE transactions on pattern analysis and machine intelligence*, vol. 23, no. 2, pp. 228–233, 2001. 3

[19] E. D. Sontag, "Vc dimension of neural networks," *NATO ASI Series F Computer and Systems Sciences*, vol. 168, pp. 69–96, 1998. 3

[20] S. Du, Y. Tao, and A. M. Martinez, "Compound facial expressions of emotion," *Proceedings of the National Academy of Sciences*, vol. 111, no. 15, pp. E1454–E1462, 2014. 4

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Computer Vision and Pattern Recognition (CVPR)*, 2015. 5, 6

[22] W. Li, F. Abtahi, Z. Zhu, and L. Yin, "Eac-net: A region-based deep enhancing and cropping approach for facial action unit detection," *arXiv preprint arXiv:1702.02925*, 2017. 6

[23] R. Zhao, Y. Wang, C. F. Benitez-Quiroz, Y. Liu, and A. M. Martinez, "Fast and precise face alignment and 3d shape reconstruction from a single 2d image," in *European Conference on Computer Vision*, pp. 590–603, Springer International Publishing, 2016. 6

[24] F. Wang, X. Xiang, C. Liu, T. D. Tran, A. Reiter, G. D. Hager, H. Quon, J. Cheng, and A. L. Yuille, "Transferring face verification nets to pain and expression regression," *arxiv*, vol. 1702, no. 06925, p. 5, 2017. 6

[25] T. Poggio and S. Smale, "The mathematics of learning: Dealing with data," *Notices of the AMS*, vol. 50, no. 5, pp. 537–544, 2003. 6

[26] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based r-cnns for fine-grained category detection," in *European Conference on Computer Vision*, pp. 834–849, Springer, 2014. 6

[27] Y. Cui, F. Zhou, Y. Lin, and S. Belongie, "Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop," *arXiv preprint arXiv:1512.05227*, 2015. 6

[28] S. Mendelson, "Improving the sample complexity using global data," *IEEE Transactions on Information Theory*, vol. 48, no. 7, pp. 1977–1991, 2002. 6

[29] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014. 6

[30] L. Ding and A. M. Martinez, "Precise detailed detection of faces and facial features," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–7, 2008. 6

[31] K. Zhao, W.-S. Chu, F. De la Torre, J. F. Cohn, and H. Zhang, "Joint patch and multi-label learning for facial action unit detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2207–2216, 2015. 6

[32] J. Zeng, W.-S. Chu, F. De la Torre, J. F. Cohn, and Z. Xiong, "Confidence preserving machine for facial action unit detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3622–3630, 2015. 6

[33] C. A. Corneanu, M. Oliu, J. F. Cohn, and S. Escalera, "Survey on rgb, 3d, thermal, and multimodal approaches for facial expression recognition: History, trends, and affect-related applications," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2016. 6