

# Deep Structure Inference Network for Facial Action Unit Recognition

Ciprian Corneanu<sup>1,2</sup>, Meysam Madadi<sup>2</sup>, and Sergio Escalera<sup>1,2</sup>

<sup>1</sup> University of Barcelona, Barcelona, Spain

<sup>2</sup> Computer Vision Center, Barcelona, Spain

{cipriancorneanu, meysam.madadi, sergio.escalera.guerrero}@gmail.com

**Abstract.** Facial expressions are combinations of basic components called Action Units (AU). Recognizing AUs is key for general facial expression analysis. Recently, efforts in automatic AU recognition have been dedicated to learning combinations of local features and to exploiting correlations between AUs. We propose a deep neural architecture that tackles both problems by combining learned local and global features in its initial stages and replicating a message passing algorithm between classes similar to a graphical model inference approach in later stages. We show that by training the model end-to-end with increased supervision we improve state-of-the-art by 5.3% and 8.2% performance on BP4D and DISFA datasets, respectively.

**Keywords:** Computer Vision, Machine Learning, Deep Learning, Facial Expression Analysis, Facial Action Units, Structure Inference

## 1 Introduction

Facial expressions (FE) are important cues for recognizing non-verbal behaviour. The ability to automatically mine human intentions, attitudes or experiences has many applications like building socially aware systems [18, 4], improving e-learning [9], adapting game status according to player’s emotions [1], and detecting deception during police interrogations [11].

The Facial Action Unit System (FACS) [5] is a descriptive coding scheme of FEs that focuses on what the face can do without assuming any cognitive or emotional value. Its basic components are called Action Units (AU) and they combine to form a complete representation of FEs.

AUs are patterns of muscular activation and the way they modify facial morphology is localized (Fig. 1a). While initial AU recognition methods (like JPML [25] and APL [28]) were using shallow predefined representations, recent methods (like DRML [26], ROI [12] and GL [7]) applied deep learning to learn richer local features that capture facial morphology. Therefore one could predict specific AUs from informative face regions selected depending on the facial geometry. For instance, contrary to non-adaptive methods like DRML [26] and APL[28], ROI [12] and JPML [25] extract features around facial landmarks which are more robust with respect to non-rigid shape changes. Patch learning is challenging as the



Fig. 1: Patch and structure learning are key problems in AU recognition. (a) By masking a region an expressive face becomes indistinguishable from neutral. (b) Multiple, correlated AUs can be active at the same time.

human face is highly articulated and different patches can contribute to either specific or groups of AUs. Learning the best patch combination together with learning specific features from each patch could be beneficial for AU recognition.

AU recognition is also multi-label. Several AUs can be active at the same time and certain AU combinations are more probable than others (Fig. 1b). AU prediction performance could be improved by considering probabilistic dependencies. In deep learning approaches, correlations can be addressed implicitly in the fully connected layers (e.g. DRML [26], GL [7] and ROI [12]). However, structure is not learned explicitly and inference and sparsity are implicit by design. JPML [25] treats the problem by including pre-learned priors about AU correlations into their learning. Learning structured outputs has also been studied by using Graphical Models [25, 19, 6]. However, these models are not end-to-end trainable.

In this work, we claim that patch and the structure learning are key problems in dealing with AU recognition. We propose a deep neural network that tackles those problems in an integrated way through an incremental and end-to-end trainable approach. First, the model learns local and holistic representations exhaustively from facial patches. Then it captures structure between patches by predicting specific AUs. Finally, AU correlations are captured by a structure inference network that replicates message passing inference algorithms in a connectionist fashion. Tab. 1 compares some of the most important features of the proposed method to the state-of-the-art (specifically JPML [25], APL [28], DRML[26], GL[7] and ROI[12]). We show that by separately treating problems in different parts of the network and being able to optimize them jointly, we improve state-of-the-art by 5.3% and 8.2% performance on BP4D and DISFA datasets, respectively. Summarizing, our 2 main contributions are: 1) we propose a model that learns representation, patch and output structure end-to-end, and 2) we introduce a structure inference topology that replicates inference algorithm in probabilistic graphical models by using a recurrent neural network.

| method    | LRL | AP | PL | SL | EE | method      | LRL | AP | PL | SL | EE |
|-----------|-----|----|----|----|----|-------------|-----|----|----|----|----|
| APL [28]  | ×   | ×  | ✓  | ×  | ×  | GL [7]      | ×   | ×  | ✓  | ×  | ✓  |
| JPML [25] | ×   | ✓  | ✓  | ×  | ×  | ROI [12]    | ✓   | ✓  | ✓  | ×  | ✓  |
| DRML [26] | ✓   | ✓  | ×  | ×  | ✓  | DSIN (ours) | ✓   | ✓  | ✓  | ✓  | ✓  |

Table 1: Features of our model and related work. LRL: local representation learning, AP: adaptive patch, PL: patch learning, SL: structured learning, EE: end-to-end.

The paper is organized as follows. Sec. 2 presents related work. Sec. 3 details the proposed model and Sec. 4 the results. Sec. 5 concludes the paper.

## 2 Related Work

Related work is discussed in relation to patch learning or structure learning.

**Patch Learning.** Inspired by locally connected convolutional layers [17], Zhao et al. [26] proposed a regional connected convolutional layer that learns specific convolutional filters from sub-areas of the input. In [12], different CNNs are trained on different parts of the face merging features in an early fusion fashion with fully connected layers. Zhao et al. [25] performed patch selection and structure learning with shallow representations where patches for each AU were selected by group sparsity learning. Jaiswal et al. [8] used domain knowledge and facial geometry to pre-select a relevant image region for a particular AU, passing it to a convolutional and bi-directional Long Short-Term Memory (LSTM) neural network. Zhong et al. [28] proposed a multi-task sparse learning framework for learning common and specific discriminative patches for different expressions. Patch location was predefined and did not take into account facial geometry.

**Structure Learning.** Zhang et al. [23] proposed a multi-task approach to learn a common kernel representation that describes AU correlations. Eleftheriadis et al. [6] adopted a latent variable Conditional Random Field (CRF) to jointly detect multiple AUs from pre-designed features. While existing methods capture local pairwise AU dependencies, Wang et al. [20] proposed a restricted Boltzmann machine that captures higher-order AU interactions. Together with patch-learning, Zhao et al. [25] used positive and negative competitions among AUs to model a discriminative multi-label classifier. Walecki et al. [19] placed a CRF on top of deep representations learned by a CNN. Both components are trained iteratively to estimate AU intensity. Wu et al. [21] used a Restricted Boltzmann Machine that captures joint probabilities between facial landmark locations and AUs. More recently, Benitez et al. [7] proposed a loss combining the recognition of isolated and groups of AUs.

## 3 Method

Let  $\mathcal{D} = \{\mathbf{X}, \mathbf{Y}\}$  be a set of pairs of input images  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_M\}$  and output AU labels  $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_M\}$  with  $M$  number of instances. Each image  $\mathbf{x}_i$

is composed of  $P$  patches  $\{I_1, \dots, I_P\}$  and output label  $\mathbf{y}_i$  is a set of  $N$  AUs  $\{y_1, \dots, y_N\}$  taking a binary value  $\{0, 1\}$ . Several AU classes can be active for an observation as a multi-label problem. Predicting such output is challenging as a softmax function can not be applied on the set of outputs contrary to the standard mono-label/multi-class problems. In addition, using independent AU activation functions in losses like cross-entropy, ignores AU correlations. Including the ability to learn structure in the model design is thus relevant.

Two main ways of solving multi-label learning in AU recognition are either capturing correlations through fully-connected layers [26, 7, 12] or inferring structure through probabilistic graphical models (PGM) [25, 19, 6]. While the former can capture correlations between classes, this is not done explicitly. On the other hand, PGMs offer an explicit solution and their optimization is well studied. Unfortunately, placing classical PGMs on top of neural network predictions considerably lowers the capacity of the model to learn high order relationships since it is not end-to-end trainable. One solution is to replicate graphical model inference in a connectionist fashion which would make possible joint optimization. Jointly training CNNs and CRFs has been previously studied in different problems [27, 2, 3]. Following this trend, in this work we formulate AU recognition by a graphical model and implement it by neural networks, more specifically CNNs and recurrent neural network (RNN). This way, AU predictions from local regions along AU correlations are learned end-to-end.

Let  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  denote a graph with vertices  $\mathcal{V} = \mathbf{y}$  specifying AUs and edges  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$  indicating the relationships between AUs. Given the Gibbs distribution we compute conditional probability  $P(\mathbf{y}|\mathbf{x}, \Theta)$  as:

$$P(\mathbf{y}|\mathbf{x}, \Theta) = \frac{1}{Z(\mathbf{y}, \mathbf{x}, \Theta)} e^{-E(\mathbf{y}|\mathbf{x}, \Theta)}, \quad (1)$$

where  $\Theta$  are model parameters,  $Z$  is a normalization function and  $E$  is an energy function. The model can be updated by introducing latent variables  $\mathbf{p}$  as:

$$P(\mathbf{y}|\mathbf{x}, \Theta) = \sum_{\mathbf{p}} P(\mathbf{y}, \mathbf{p}|\mathbf{x}, \Theta), \quad (2)$$

where  $\mathbf{p}$  is given as the output of CNN. The vertices and edges in the graph  $\mathcal{G}$  can be updated as  $\mathcal{V} = \mathbf{y} \cup \mathbf{p}$  and  $\mathcal{E} = \mathcal{E}_y \cup \mathcal{E}_{py} \cup \mathcal{E}_p$ . Although edges  $\mathcal{E}_y$  can be defined by a prior knowledge taken from a given dataset, we use a fully connected graph independent to the dataset and assign a mutual gating strategy to control information passing through edges (more details in Sec. 3.3). We define  $\mathcal{E}_{py}$  as edges between  $\mathbf{p}$  and  $\mathbf{y}$ , and use a selective strategy to define edges in this set. Finally, edges  $\mathcal{E}_p$  is an empty set, since in our model an independent CNN is trained on each image patch  $I_j$  and we do not assign any edge among  $\mathbf{p}$ . Given this assumption, probability distribution  $P(\mathbf{y}, \mathbf{p}|\mathbf{x}, \Theta)$  is given by:

$$P(\mathbf{y}, \mathbf{p}|\mathbf{x}, \Theta) = P(\mathbf{y}|\mathbf{p}, \mathbf{x}, \Theta) \prod_k P(p_k|\mathbf{x}, \Theta). \quad (3)$$

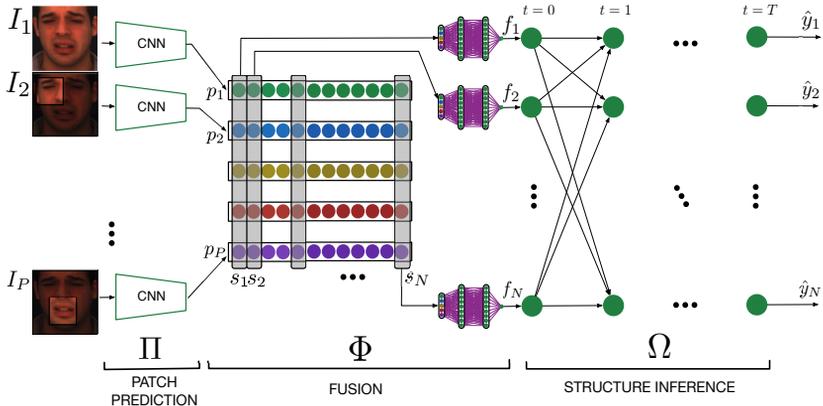


Fig. 2: Deep Structure Inference Network (DSIN) learns independent AU predictions from global and local learned features. It refines each AU prediction by taking into account correlation to the other AUs. Each input image is cropped into a set of patches  $\{I_i\}_{i=1}^P$  which is used for training an independent CNN for producing a probability vector  $p_i$  for  $N$  AUs ( $\varphi_p$  in Eq. 4). From  $s_j$  (the patch predictions for a specific AU) we learn a combination for producing a single AU prediction  $f_j$  (simplified  $\psi_{py}$  in Eq. 4). Final predictions  $y_j$  are computed by inferring structure among AUs through iterative message passing similar to inference in a probabilistic graph model ( $\psi_y$  in Eq. 4).

As in CRF, energy function  $E(\cdot)$  is computed by unary and pairwise terms as:

$$E(\mathbf{y}, \mathbf{p}, \mathbf{x}, \Theta) = \sum_k \varphi_p(p_k, \mathbf{x}, \pi) + \sum_{(i,k) \in \mathcal{E}_{py}} \psi_{py}(y_i, p_k, \phi) + \sum_{(i,j) \in \mathcal{E}_y} \psi_y(y_i, y_j, \omega), \quad (4)$$

where  $\varphi(\cdot)$  is a unary term,  $\psi_*(\cdot)$  are pairwise terms and  $\Theta = \pi \cup \phi \cup \omega$ . Fig. 2 presents our Deep Structure Inference Network (DSIN). It consists of three components each designed to solve a term in Eq. 4. We refer to the initial part as *Patch Prediction* (PP), whose purpose is to exhaustively learn deep local representations from facial patches and produce local predictions. Then, the *Fusion* (F) module performs patch learning per AU. The final stage, *Structure Inference* (SI), refines AU prediction by capturing relationships between AUs. The DSIN is end-to-end trainable and CNN features can be trained based on gradients back-propagated from structure inference in a multi-task learning fashion.

### 3.1 Patch Prediction

Given image patches  $\mathbf{x}$ , unary terms  $\varphi_p(\mathbf{p}, \mathbf{x}, \pi)$  provide AUs confidences for each patch which are defined as the log probability:

$$\varphi_p(\mathbf{p}, \mathbf{x}, \pi) = \log P(\mathbf{p}|\mathbf{x}, \pi). \quad (5)$$

Probability  $P(\mathbf{p}|\mathbf{x}, \pi)$  is modeled by independent patch prediction functions  $\{\Pi_i(I_i; \pi_i)\}_{i=1}^P$ , where  $I_i$  is input image patch and  $\pi_i$  are function parameters.

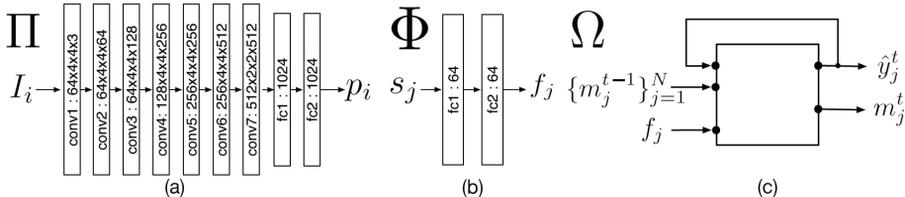


Fig. 3: (a) Topology of patch prediction CNNs. Each convolutional block has stride 2 and batch normalization. Number of filters followed by the size of the kernel are marked. The last layers are fully-connected (FC) layers marked with the number of neurons. All neurons use ReLU activations. (b) Each fusion unit is a stack of 2 FC layers. (c) A structure inference unit. For better visualization, we just show the interface of the unit without the inner topology. See details in Sec. 3.3.

Each  $\Pi_i$  is a CNN computing  $N$  AUs probabilities through sigmoid function at last layer.  $P$  independent predictions are provided at this stage, each being a vector of AU predictions. Although image patches may overlap, we assume independence to let each network be expert at predicting AUs on local regions. By learning independent global representations and local representations, we can better capture facial morphology and address AU locality.

In Fig. 3(a) we detail the topology of the CNNs used for learning the patch prediction functions. Many complex topologies have been proposed in recent years and searching for the best is out of the scope of this work. The chosen topology, a shallow network, follows the intuition behind well known models like VGG [16].

### 3.2 Fusion

Computational complexity to marginalize pairwise relationships in  $\mathcal{E}_{py}$  is high. In our formulation, we simplify edges such that  $\mathcal{E}_{py}$  becomes directed from nodes in  $\mathbf{p}$  to nodes in  $\mathbf{y}$ . It means we omit mutual relationships among  $\mathbf{p}$  and  $\mathbf{y}$ . Therefore, nodes in  $\mathbf{y}$  are conditioned on the nodes in  $\mathbf{p}$ . However, we want each AU node in  $\mathbf{y}$  to be conditioned on the same AU nodes in  $\mathbf{p}$  from different patches. It means different patches can provide complementary information to predict target AU independent to other AUs. Finally,  $\psi_{py}(\mathbf{y}, \mathbf{p}, \phi)$  is defined as the log probability of  $P(\mathbf{y}|\mathbf{p}, \phi)$  which is modeled by a set of independent functions, so called fusion functions  $\{\Phi_j(s_j; \phi_j)\}_{j=1}^N$ , where  $s_j \subset \mathbf{p}$  corresponds to the set of  $j$ -th AU predictions from all patches and  $\phi_j$  is function parameters. We simply model each function  $\Phi_j$  with 2 fully connected layers with 64 hidden units, each followed by a sigmoid layer, as shown in Fig. 3(b). We found 64 hidden units works well in practice while higher dimensionality does not bring any additional performance and quickly starts over-fitting. The output of each  $\Phi_j$  is the predicted probability  $f_j$  for  $j$ -th AU.

### 3.3 Structure Inference

Up to now, we computed individual AU probabilities in a feed-forward neural network without taking AU relationships explicitly into account. The goal is to model pairwise terms  $\psi_y$  such that the whole process is end-to-end trainable in a compact way. Belief propagation and message passing between nodes is one of the well known algorithms for PGM inference. Inspired by [3], which proposes a connectionist implementation for action recognition, we build a *Structure Inference* (SI) module in the final part of DSIN.

The SI updates each AU prediction in an iterative manner by taking into account information from other AUs. The intuition behind this is that by passing information between predictions in an explicit way, we can capture AU correlations and improve predictions. The structure inference module is a collection of interconnected recurrent structure interference units (SIU) (see Fig. 3(c)). For each AU there is a dedicated SIU. We denote the computations done by SIU by a function  $\Omega$ . Let  $\{\Omega_j\}_{j=1}^N$  be the set of SIU functions  $\Omega_j : \mathbb{R}^{N+2} \rightarrow \mathbb{R}^2$  where:

$$\hat{y}_j^t, m_j^t = \Omega_j(f_j, m_1^{t-1}, m_2^{t-1}, \dots, m_N^{t-1}, \hat{y}_j^{t-1}; \omega_j). \quad (6)$$

At each iteration  $t$ ,  $\Omega_j$  takes as input the initial prediction  $f_j$  for its class, a set of incoming messages  $\{m_j^{t-1}\}_{j=1}^N$  from the SIUs corresponding to the other classes and its own previous prediction  $\hat{y}_j^{t-1}$ . Each function  $\Omega_j$  has two inline units: producing  $j$ -th AU prediction  $\hat{y}_j^t$  and message  $m_j^t$  for next time step. In this way, predictions are improved iteratively by receiving information from other nodes. Computationally, we replicate this iterative message passing mechanism in the collection of SIUs with a recurrent neural network that shares function parameters  $\Omega_j$  across all time steps. We show a SIU unit in Fig. 3(c).

A message unit basically corresponds to the distribution of the AU node. A message unit from a SIU is a parametrized function of the previous messages, the initial fused prediction and the previous prediction of the same SIU:

$$m_j^t = \sigma(\omega_j^m [\mu(m_1^{t-1}, \dots, m_N^{t-1}), f_j, \hat{y}_j^{t-1}] + \beta_j^m), \quad (7)$$

where  $\sigma(\cdot)$  is the sigmoid function,  $\mu(\cdot)$  is the mean function,  $\omega_j^m \in \mathbb{R}^3$  and  $\beta_j^m \in \mathbb{R}$  are message function parameters. Messages between two nodes at each time step have a mutual relationship which can be controlled by a gating strategy. Therefore, a set of correction factors are computed as:

$$\chi_j^t = \sigma(\omega_j^g [\mu(m_1^t, \dots, m_N^t), f_j, \hat{y}_j^{t-1}] + \beta_j^g), \quad (8)$$

where  $\omega_j^g \in \mathbb{R}^3$  and  $\beta_j^g \in \mathbb{R}$  are gating function parameters. Then, a message  $m_{i \rightarrow j}^t$  that is passed from AU node  $i$  to  $j$  will be updated by the mutual factors of the gate between nodes  $i$  and  $j$  as:

$$\bar{m}_j^t = \mu(\chi_i^t, \chi_j^t) m_{i \rightarrow j}^t. \quad (9)$$

Finally, updated messages coming to the  $j$ -th node along with initial estimation  $f_j$  are used to produce output prediction  $\hat{y}_j^t$  as:

$$\hat{y}_j^t = \sigma(\omega_j^y [\mu(\bar{m}_1^t, \dots, \bar{m}_N^t), f_j] + \beta_j^y), \quad (10)$$

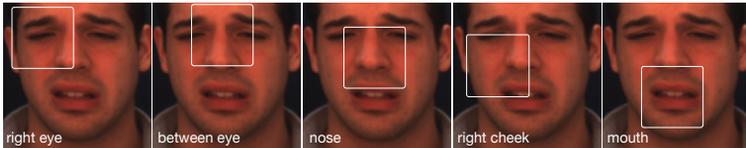


Fig. 4: Each input image is aligned and cropped into 5 patches.

where  $\omega_j^y \in \mathbb{R}^2$  and  $\beta_j^y \in \mathbb{R}$  are prediction function parameters. By doing this, we are able to combine representation learning in function  $\Pi$ , patch learning in function  $\Phi$  and structure inference in the  $\Omega$  in a single end-to-end trainable model. We introduce our training strategy in Sec. 4.1.

## 4 Experimental Analysis

In the following, we describe experimental settings and results.

### 4.1 Experimental Setting

**Data.** We used BP4D [24] and DISFA [13] datasets. BP4D contains 2D and 3D videos of 41 young adults. It has 328 videos (8 videos for 41 participants) with 12 coded AUs, resulting in about 140k valid face images [24]. DISFA contains 27 adults (12 women and 15 men) with ages between 18 to 50 years and relative ethnic diversity. The data corpus consists of approximately 130k frames in total. AU intensity is coded for each video frame on a 0 (not present) to 5 (maximum intensity) ordinal scale. For our purpose we consider all labels with intensity greater than 3 as active and the rest as non-active. Both datasets are widely used in most recent AU recognition works.

**Preprocessing.** For each image, facial geometry is estimated using [10]. From all neutral faces we compute 3 reference anchors as the mean of the eyes and the mouth landmarks. Faces are resized to  $224 \times 224 \times 3$  and a rigid transformation is applied for registering to the anchors, reducing variance to scale and rotation. We crop 5 patches of size  $56 \times 56 \times 3$  around points defined by the detected landmarks (see Fig. 4). For reducing redundancy we ignore corresponding, symmetrical patches like the left eye and cheek.

**Training.** We incrementally train each part of DSIN before end-to-end model training. During training we use supervision on the patch prediction  $p$ , the fusion  $f$  and the structure inference outputs  $\hat{y}$ . On  $p$  we use a weighted  $L_2$  loss denoted by  $L_\Pi(p, y)$ . The weights are inversely proportional to the ratio of positives in the total number of observations for each AU class in training. The weighting gives more importance to the minority classes in each training batch which ensures a more equal gradient update across classes and overall better performance. On the fusion and structure inference outputs we apply a binary cross-entropy loss (denoted by  $L_\Phi(f, y)$  and  $L_\Omega(\hat{y}, y)$ ). For the structure inference we include

**Algorithm 1:** Training procedure of DSIN.**Training data:**  $\{\{I\}_{i=1}^P, y\}$ **Model parameters:** patch prediction:  $\{\pi_i\}_{i=1}^P$ , fusion  $\{\phi_i\}_{i=1}^N$ , structure inference  $\{\omega_i\}_{i=1}^N$ **Step 0:** random initialization around 0:  $\pi, \phi, \omega \leftarrow \mathcal{N}(0, \sigma^2)$ **Step 1:** train patch prediction:  $\pi_i \leftarrow \min_{\pi} (L_{\Pi}(\Pi_i(I_i; \pi_i)), y), \forall i \in \{1, \dots, P\}$ **Step 2:** freeze patch prediction; train fusion:  $\phi \leftarrow \min_{\phi} L_{\Phi}(\Phi(\Pi; \phi), y)$ **Step 3:** train patch prediction and fusion jointly:

$$\pi, \phi \leftarrow \min_{\pi, \phi} (L_{\Pi}(\Pi(I; \pi)), y) + L_{\Phi}(\Phi(\Pi; \phi), y)$$

**Step 4:** freeze patch prediction and fusion; train structure inference:

$$\omega \leftarrow \min_{\omega} L_{\Omega}(\Omega(\Phi; \omega), y)$$

**Step 5.** train all:

$$\pi, \phi, \omega \leftarrow \min_{\pi, \phi, \omega} (w_1 L_{\Pi}(\Pi(I; \pi)), y) + w_2 L_{\Phi}(\Phi(\Pi; \phi), y) + w_3 L_{\Omega}(\Omega(\Phi; \omega), y))$$

**Output:** optimized parameter:  $\pi^{opt}, \phi^{opt}, \omega^{opt}$ 

a regularization on the correction factors (denoted by  $\chi$  in Eq. 8 and Eq. 9) to force sparsity in the message passing. Details of the training procedure are shown in Alg. 1. We use an Adam optimizer with learning rate of 0.001 and mini-batch size 64 with early stopping. Experimentally, we found the individual loss contributions  $w_1 = 0.25$ ,  $w_2 = 0.25$  and  $w_3 = 0.5$  to work well in training. For both datasets we perform a subject exclusive 3-fold cross-validation. Similarly to [12], on DISFA we take the best CNNs trained for patch prediction on the BP4D and retrained fully connected layers for the new set of outputs. We fix the convolutional filters throughout the rest of the training.

**Methods and metrics.** We compare against CPM [22], APL [28], JPML [25], DRML [26], and ROI [12] state-of-the-art alternatives. We evaluate F1-frame score as  $F1 = 2 \frac{PR}{P+R}$ , where  $P = \frac{tp}{tp+fp}$ ,  $R = \frac{tp}{tp+fn}$ ,  $tp$  being true positives,  $fn$  false negatives and  $fp$  false positives. All metrics are computed per AU and then averaged. Targeted AUs shown in Fig. 6.

## 4.2 Results

In the following, we explore the effect design decisions included in the DSIN followed by comparison against state-of-the-art alternatives in Sec. 4.2 and qualitative examples in Sec. 4.2.

**Ablation Study.** We analyze DSIN design decisions in the following.

**Class balancing.** In both datasets, classes are strongly imbalanced. This can be harmful during training. To alleviate this, we use a weighted loss on patch prediction CNNs. Tab. 2 shows results with and without class balancing. This overall improves performance, especially on poorly represented classes. On BP4D the classes with ratios of positives in the total of samples lower than 30% are AU01, AU02, AU04, AU17, AU24. These are the classes that are improved the most. AUs like AU07 or AU12 have positives to total ratios higher than 50%. Balancing can reduce performance on these classes.

|      | method                           | AU01        | AU02        | AU04        | AU06        | AU07        | AU10        | AU12        | AU14        | AU15        | AU17        | AU23        | AU24        | avg         |
|------|----------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|      | VGG(face) <sup>ft</sup>          | <b>35.2</b> | <b>31.2</b> | 25.4        | <b>73.1</b> | <b>72.1</b> | 80.1        | 59.2        | 35.1        | 32.1        | 52.3        | 26.1        | <b>36.2</b> | 46.5        |
|      | PP(face) <sup>ncb</sup>          | 35.1        | <b>38.1</b> | <b>53.9</b> | <b>77.2</b> | 70.7        | <b>83.1</b> | <b>86.2</b> | <b>56.1</b> | <b>39.8</b> | <b>54.5</b> | <b>37.2</b> | 31.4        | <b>55.3</b> |
| PP   | PP(right eye) <sup>ind</sup>     | <b>46.8</b> | <b>40.4</b> | 45.3        | 68.3        | 69.2        | -           | -           | -           | -           | -           | -           | -           | -           |
|      | PP(mouth) <sup>ind</sup>         | -           | -           | -           | -           | -           | 78.6        | 82.0        | 54.2        | 38.6        | 54.7        | [39.3]      | <b>43.3</b> | -           |
|      | PP(right eye)                    | 38.0        | [37.7]      | 48.3        | 69.5        | 71.0        | 72.4        | 77.4        | 50.7        | 15.0        | 38.9        | 13.8        | 15.3        | 45.7        |
|      | PP(between eye)                  | 41.7        | 34.8        | 45.9        | 64.9        | 65.5        | 72.1        | 73.9        | 54.9        | 19.7        | 33.9        | 13.9        | 7.0         | 44.0        |
|      | PP(mouth)                        | 12.4        | 7.3         | 22.4        | 75.5        | 70.5        | 78.9        | 81.3        | <b>66.2</b> | 35.8        | 59.6        | 37.6        | [42.8]      | 49.3        |
|      | PP(right cheek)                  | 30.5        | 18.4        | 41.8        | 75.2        | 73.2        | 79.1        | 81.9        | [61.9]      | 35.7        | 55.1        | 35.5        | 35.7        | 52.0        |
|      | PP(nose)                         | 41.6        | 28.4        | 46.4        | 71.1        | 70.5        | 78.8        | 78.0        | 57.1        | 21.3        | 43.7        | 34.0        | 20.3        | 49.3        |
|      | PP(face)                         | 43.8        | 37.5        | [54.9]      | <b>77.4</b> | [71.2]      | [79.2]      | <b>84.0</b> | 56.6        | [39.7]      | [59.7]      | 39.2        | 39.5        | [56.9]      |
|      | PP+F                             | [44.8]      | 35.8        | <b>57.1</b> | [76.7]      | <b>74.3</b> | <b>79.6</b> | [83.7]      | 56.6        | <b>41.1</b> | <b>61.8</b> | <b>42.2</b> | 40.1        | <b>57.8</b> |
| DSIN | DSIN <sub>2</sub> <sup>ncf</sup> | 46.7        | 34.1        | <b>62.0</b> | 76.5        | <b>74.1</b> | [83.1]      | 84.9        | 60.9        | 36.0        | 57.1        | <b>43.3</b> | 36.1        | 57.9        |
|      | DSIN <sub>2</sub>                | 47.7        | 36.5        | 55.6        | 76.3        | [73.7]      | 80.1        | 85.0        | 64.0        | [39.2]      | 60.6        | [43.1]      | 39.9        | 58.2        |
|      | DSIN <sub>5</sub>                | [49.7]      | 36.3        | 57.3        | <b>76.8</b> | 73.4        | 81.6        | 84.5        | [64.7]      | 38.5        | [63.0]      | 39.0        | 37.3        | 58.5        |
|      | DSIN <sub>10</sub>               | <b>51.7</b> | [40.4]      | 56.0        | 76.1        | 73.5        | 79.9        | [85.4]      | 62.7        | 37.3        | 62.9        | 38.6        | [41.6]      | [58.9]      |
|      | DSIN <sub>10</sub> <sup>f</sup>  | <b>51.7</b> | <b>41.6</b> | [58.1]      | [76.6]      | <b>74.1</b> | <b>85.5</b> | <b>87.4</b> | <b>72.6</b> | <b>40.4</b> | <b>66.5</b> | 38.6        | <b>46.9</b> | <b>61.7</b> |

Table 2: Recognition results on BP4D. PP([patch]) stands for patch prediction on the indicated patch. F stands for the fusion and DSIN is the final model. We indicate the results when training on individual AUs with [method]<sup>ind</sup>, fine tuning on the validation dataset of the decision threshold by DSIN<sup>tt</sup>, number of iterations of the structure inference by DSIN<sub>T</sub> and training without correction factors as DSIN<sup>ncf</sup>. VGG(face)<sup>ft</sup> is a pre-trained VGG-16 [14] fine-tuned on BP4D. PP(face)<sup>ncb</sup> is a patch prediction without class balancing. All results are obtained by 3-fold cross-validation on BP4D.

**Choice of prediction topology.** In Tab. 2 we compare the proposed CNN for patch prediction (PP(face)) against VGG-16. The VGG-16 model used was trained for face recognition [14] and fine-tuned on our data for AU recognition. Our model shows superior performance.

**Targeting subsets of AUs.** We explore the effect of the considered target set on the overall prediction performance. In Tab. 2 we show prediction results from the right eye and from the mouth patches when training either on the full set of targets ([method]) or on individual targets ([method]<sup>ind</sup>). When training on individual AUs the decision for the classifier is simpler. On the other hand any correlation information between classes that could be captured by the FC layers is ignored. In certain cases the individual prediction is superior to the exhaustive prediction. In the case of the right eye patch this is particularly true for AU01. But this is rather the exception. On average and across patches training on groups of AUs or on all AUs is beneficial as correlation information between classes is employed by the network in the fully connected layers. Additionally, predicting AU individually with independent nets would quickly increase the number of parameters with considerable effects on the training speed and final model performance.

Tab. 2 and 3 show AU recognition results on both datasets trained on patches. That proves the locality assumption. When training on the mouth the performance on the upper face AUs is greatly affected. Similarly, training on the eye affects the performance on the lower face AUs. This is expected as the patch prediction can only infer the other AUs from the ones visible in the patch.

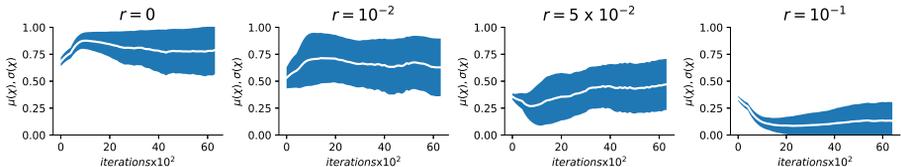


Fig. 5: Different levels of regularization on the mean  $\mu(\chi)$  (white line) and standard deviation  $\sigma(\chi)$  (envelope) of the correction factors during training. Small regularization values force the correction factors to diverge faster. Increasing regularization collapses the correction factors hurting the message passing.

| method          | AU01        | AU02        | AU04        | AU06        | AU09        | AU12        | AU25        | AU26        | avg         |
|-----------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| PP(right eye)   | 27.2        | 15.4        | 58.8        | 8.0         | 18.2        | 53.6        | 73.3        | 9.1         | 33.0        |
| PP(between eye) | 34.6        | 13.2        | 59.7        | 15.4        | 21.1        | 50.9        | 72.9        | 8.5         | 34.5        |
| PP(mouth)       | 7.5         | 6.4         | 44.6        | 28.5        | 23.9        | <b>72.1</b> | 87.5        | [27.3]      | 37.2        |
| PP(right cheek) | 24.6        | 12.2        | 46.1        | 31.2        | 45.2        | 71.5        | 84.5        | 22.4        | 33.8        |
| PP(nose)        | 21.9        | 19.1        | 52.0        | <b>32.0</b> | <b>50.9</b> | 66.5        | 76.6        | 8.9         | 41.0        |
| PP(face)        | 29.8        | [31.4]      | 64.6        | 26.8        | 21.3        | 70.1        | 87.0        | 20.3        | 43.9        |
| PP+F            | [40.1]      | 18.6        | <b>70.8</b> | 25.4        | 42.1        | [71.8]      | [88.8]      | 26.4        | [48.0]      |
| DSIN            | <b>42.4</b> | <b>39.0</b> | [68.4]      | [28.6]      | [46.8]      | 70.8        | <b>90.4</b> | <b>42.2</b> | <b>53.6</b> |

Table 3: Results of DSIN on DISFA. PP([patch]) stands for patch prediction on the indicated patch. F stands for the fusion. DSIN is the final model. For DISFA we only show the DSIN with  $T = 10$ , the best performing on BP4D.

**Learning Local Representations.** On average, face prediction compared to patch prediction performs better on the entire output set. However, when individual AUs are considered, this is no longer the case. For BP4D, the performance on AU15 and AU24 are considerably higher when predicting from the mouth patch than from the face (see Tab. 2). On DISFA the prediction from the whole face is the best on just 3 AUs (see Tab. 3). The nose patch is better for predicting AU06 and AU09, the mouth patch is better for AU12, AU25 and AU26, and the between eye patch for AU01.

**Patch Learning.** Tab. 2 and 3 show results of AU-wise fusion for BP4D and DISFA (PP+F). On both, patch learning through fusion is beneficial, but on DISFA benefits are higher. This might be due to the fact that prediction results on DISFA are considerably more balanced across patches. Overall on BP4D the fusion improves results on almost all AUs compared to face prediction. This shows that even though the other patches perform worse on certain classes, there is structure to learn from their prediction that helps to improve performance. However, the fusion is not capable to replicate the result of the mouth prediction on AU14. On DISFA, in almost every case fusion gets close or higher to the best patch prediction. In both cases, fusion has greater problems in improving individual patches in cases where input predictions are already very noisy.

**Structure Learning.** Tab. 2 and 3 show results of the final DSIN model. For BP4D, we also perform a study of the number of iterations  $T$  considered for structure inference. Since parameters  $\omega_j$  are shared across iterations, more iterations are beneficial to capture AU relationships in a fully connected graph with a



Fig. 6: Facial Action Units targeted in this work.

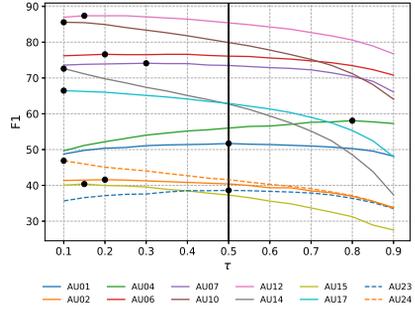


Fig. 7:  $\tau$  vs AU performance on BP4D validation set. Black circles denote best score.

large number of nodes (12 in our case). We also trained DSIN without correction factors (Eq. 9 is not applied in this case). Results are inferior compared with the same model with correction factors. In the case of DISFA, we only applied the structure inference with the best previously found  $T = 10$  steps. Structure inference is beneficial in both cases. On BP4D, it considerably improves AU2 and AU14. For DISFA, the results are even more conclusive. Adding the structure inference brings more than 5% improvement over the fusion.

**Correction factor regularization.** Fig. 5 shows the effect of increasing regularization applied on the correction factors  $\chi$ . Overall, regularizing  $\chi$  does not bring significant benefits. When comparing  $r = 10^{-2}$  with no regularization the differences are minimal. The network has the ability to learn sparse message passing by itself without regularization. Still, small values of  $r$  lead to faster divergence of  $\chi$  and faster convergence of the network. The difference in performance is not significant. On the other hand values of  $r > 5 \times 10^{-2}$  negatively affect performance as most of  $\chi$  get closer to 0 and no messages are passed anymore. For these reasons, we keep  $r = 5 \times 10^{-3}$ .

**Threshold Tuning.** Prediction value per AU takes values between 0 and 1. In all results, we compute the performance by binarizing the output with respect to threshold  $\tau = 0.5$ . Although class balancing as a weighted loss is beneficiary, it does not totally solve data imbalance. Fig. 7 shows performance in terms of  $\tau$  for validation set of BP4D. As shown, a threshold  $\tau = 0.5$  is not an ideal value. For most classes  $\tau \in [0.1, 0.3]$  is preferable. Exception is AU04. Tables 2 and 3 show the performance of the proposed model after tuning  $\tau$  per class (DSIN<sup>tt</sup>). This way 2.8% and 3.1% of performance is gained on BP4D and DISFA, respectively.

**Comparison with state-of-the-art.** Tables 4 and 5 show how our model compares against the state-of-the-art related methods on BP4D and DISFA, respectively. DSIN and ROI are the best performing in both datasets. Both methods learn deep local representations and patch combinations end-to-end. The worst performing methods, JPML on BP4D and APL on DISFA, use pre-defined features and are not end-to-end trained. Comparing DSIN and ROI with

| method             | AU01        | AU02        | AU04        | AU06        | AU07        | AU10        | AU12        | AU14        | AU15        | AU17        | AU23        | AU24        | AVG         |
|--------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| JPML [25]          | 32.6        | 25.6        | 37.4        | 42.3        | 50.5        | 72.2        | 74.1        | [65.7]      | 38.1        | 40.0        | 30.4        | [42.3]      | 45.9        |
| DRML [26]          | 36.4        | <b>41.8</b> | 43.0        | 55.0        | 67.0        | 66.3        | 65.8        | 54.1        | 33.2        | 48.0        | 31.7        | 30.0        | 48.3        |
| CPM [22]           | [43.4]      | 40.7        | 43.3        | 59.2        | 61.3        | 62.1        | 68.5        | 52.5        | 36.7        | 54.3        | <b>39.5</b> | 37.8        | 50.0        |
| ROI [12]           | 36.2        | 31.6        | 43.4        | <b>77.1</b> | [73.7]      | [85.0]      | [87.0]      | 62.6        | <b>45.7</b> | 58.0        | 38.3        | 37.4        | 56.4        |
| DSIN               | <b>51.7</b> | 40.4        | [56.0]      | 76.1        | 73.5        | 79.9        | 85.4        | 62.7        | 37.3        | [62.9]      | [38.8]      | 41.6        | [58.9]      |
| DSIN <sup>tt</sup> | <b>51.7</b> | [41.6]      | <b>58.1</b> | [76.6]      | <b>74.1</b> | <b>85.5</b> | <b>87.4</b> | <b>72.6</b> | [40.4]      | <b>66.5</b> | 38.6        | <b>46.9</b> | <b>61.7</b> |

Table 4: AU recognition results on BP4D. Best results are shown in bold. Second best results are shown in brackets. For the proposed model we show an additional set of results (DSIN<sub>tt</sub>) obtained when the decision threshold is tuned per AU.

| method             | AU01        | AU02        | AU04        | AU06        | AU09        | AU12        | AU25        | AU26        | avg         |
|--------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| APL[28]            | 11.4        | 12.0        | 30.1        | 12.4        | 10.1        | 65.9        | 21.4        | 26.0        | 23.8        |
| DRML [26]          | 17.3        | 17.7        | 37.4        | 29.0        | 10.7        | 37.7        | 38.5        | 20.1        | 26.7        |
| ROI [12]           | 41.5        | 26.4        | 66.4        | <b>50.7</b> | 8.5         | <b>89.3</b> | 88.9        | 15.6        | 48.5        |
| DSIN               | [42.4]      | [39.0]      | [68.4]      | 28.6        | [46.8]      | 70.8        | [90.4]      | [42.2]      | [53.6]      |
| DSIN <sup>tt</sup> | <b>46.9</b> | <b>42.5</b> | <b>68.8</b> | [32.0]      | <b>51.8</b> | [73.1]      | <b>91.9</b> | <b>46.6</b> | <b>56.7</b> |

Table 5: AU recognition results on DISFA. Best results are shown in bold. Second best results are shown in brackets.

DRML one can observe the advantage in learning independent local representation. Both ROI and our model learn independent local representations, while DRML disentangles the representation learning in just one layer of their network. Interestingly though, there is also an exception. On BP4D, CPM performs slightly better than DRML even though it is not a deep learning method. When comparing our proposed model with ROI on BP4D our CNN trained just on face without class balancing has inferior results. When we include class balancing and patch learning our topology improves performance, further enhanced by structure inference and end-to-end final training. In the case of DISFA, single CNN trained on the whole face with class balancing has a performance of 43.9, being 4.6% lower than ROI. When we add patch prediction fusion (PP+F) we get just 0.5% lower than ROI while the addition of the structure inference and threshold tuning improves ROI performance. Finally, DSIN shows the best results on both datasets. For BP4D, from the 12 AUs target it performs best on 5 and second best on additional 5. In the case of DISFA the improvement over ROI is greater, DSIN performing best in all but one AU. Overall, we obtain 5.3% absolute and 9.4% relative performance improvement on BP4D and 8.2% absolute and 16.9% relative performance improvement on DISFA, respectively.

**Qualitative results.** Fig. 8(a) shows examples of how structure inference tends to correct predictions following AU correlations. We show the magnitude of AU correlations on BP4D in Fig. 8(b). In the first 3 column examples, AU06 and AU07 are not correctly classified by the fusion model (middle row). Both these AUs are highly correlated with already detected AUs like AU10, AU12 and AU14. Such correlation could be captured by SI (bottom row). The rightmost example shows how AU17, a false positive, is corrected. As shown in Fig. 8(b), AU17

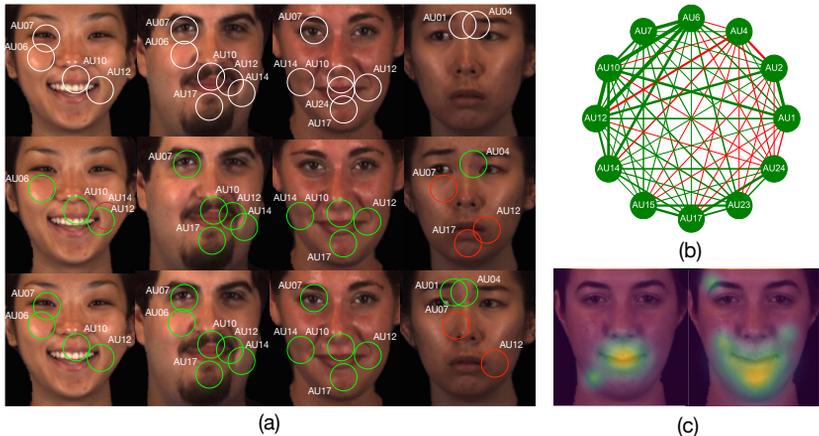


Fig. 8: (a) Examples of AU predictions: ground-truth (top), fusion module (middle) and structure inference (bottom) prediction (●: true positive, ●: false positive). (b) AUs correlation in BP4D (●: positive, ●: negative). Line thickness is proportional with correlation magnitude. (c) Class activation map for AU24 that shows the discriminative regions of simple patch prediction (left) and DSIN (right). Best seen in color.

is negatively correlated with AU4, which was already detected. In Fig. 8(c) we show a class activation map [15] for AU24 of the patch prediction (left) vs. the DSIN (right). Contrary to very localized patch prediction, the attention on right expands to a larger area of the face where possible correlated AUs might exist.

## 5 Conclusion

We proposed the Deep Structured Inference Network, designed to deal with both patch and structure learning for AU recognition. DSIN first learns independent local and global representations and corresponding predictions. Then, it learns relationships between predictions per AU through stacked fully connected layers. Finally, inspired by inference algorithms in graphical models, DSIN replicates a message passing mechanism in a connectionist fashion. This adds the ability to capture correlations in the output space. The model is end-to-end trainable and improves state-of-the-art results by 5.3% and 8.2% performance on BP4D and DISFA datasets, respectively. Future work includes learning patch structure at feature level and a structure inference module with increased capacity.

## Acknowledgements

This work has been partially supported by the Spanish project TIN2016-74946-P (MINECO/FEDER, UE) and CERCA Programme / Generalitat de Catalunya. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPU used for this research.

## References

1. Bakkes, S., Tan, C.T., Pisan, Y.: Personalised gaming. *JCT* **3** (2012)
2. Chu, X., Ouyang, W., Wang, X., et al.: Crf-cnn: Modeling structured information in human pose estimation. In: *Advances in Neural Information Processing Systems*. pp. 316–324 (2016)
3. Deng, Z., Vahdat, A., Hu, H., Mori, G.: Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition. In: *IEEE CVPR*. pp. 4772–4781 (2016)
4. DeVault, D., Artstein, R., Benn, G., Dey, T., Fast, E., Gainer, A., Morency, L.P.: A virtual human interviewer for healthcare decision support. *AAMAS* (2014)
5. Ekman, P., Friesen, W., Hager, J.: *Facs manual. a human face* (2002)
6. Eleftheriadis, S., Rudovic, O., Pantic, M.: Multi-conditional latent variable model for joint facial action unit detection. In: *IEEE ICCV*. pp. 3792–3800 (2015)
7. Fabian Benitez-Quiroz, C., Wang, Y., Martinez, A.M.: Recognition of action units in the wild with deep nets and a new global-local loss. In: *IEEE ICCV (Oct 2017)*
8. Jaiswal, S., Valstar, M.: Deep learning the dynamic appearance and shape of facial action units. In: *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*. pp. 1–8. *IEEE* (2016)
9. Kapoor, A., Burleson, W., Picard, R.W.: Automatic prediction of frustration. *IJHCS* **65**(8), 724–736 (2007)
10. Kazemi, V., Sullivan, J.: One millisecond face alignment with an ensemble of regression trees. In: *IEEE CVPR*. pp. 1867–1874 (2014)
11. Kulkarni, K., Corneanu, C.A., Ofodile, I., Escalera, S., Baro, X., Hyniewska, S., Allik, J., Anbarjafari, G.: Automatic recognition of deceptive facial expressions of emotion. *arXiv preprint arXiv:1707.04061* (2017)
12. Li, W., Abtahi, F., Zhu, Z.: Action unit detection with region adaptation, multi-labeling learning and optimal temporal fusing. In: *IEEE CVPR*. pp. 6766–6775 (2017)
13. Mavadati, S.M., Mahoor, M.H., Bartlett, K., Trinh, P., Cohn, J.F.: Disfa: A spontaneous facial action intensity database. *IEEE Transactions on Affective Computing* **4**(2), 151–160 (2013)
14. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: *British Machine Vision Conference* (2015)
15. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. See <https://arxiv.org/abs/1610.02391> v3 **7**(8) (2016)
16. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014)
17. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In: *IEEE CVPR*. pp. 1701–1708 (2014)
18. Vinciarelli, A., Pantic, M., Bourlard, H.: Social signal processing: Survey of an emerging domain. *IVC* **27**(12), 1743–1759 (2009)
19. Walecki, R., Rudovic, O., Pavlovic, V., Schuller, B., Pantic, M.: Deep structured learning for facial action unit intensity estimation. In: *IEEE CVPR*. pp. 5709–5718 (2017)
20. Wang, Z., Li, Y., Wang, S., Ji, Q.: Capturing global semantic relationships for facial action unit recognition. In: *IEEE ICCV*. pp. 3304–3311 (2013)
21. Wu, Y., Ji, Q.: Constrained joint cascade regression framework for simultaneous facial action unit recognition and facial landmark detection. In: *IEEE CVPR*. pp. 3400–3408 (2016)

22. Zeng, J., Chu, W.S., De la Torre, F., Cohn, J.F., Xiong, Z.: Confidence preserving machine for facial action unit detection. In: IEEE ICCV. pp. 3622–3630 (2015)
23. Zhang, X., Mahoor, M.H.: Task-dependent multi-task multiple kernel learning for facial action unit detection. *Pattern Recognition* **51**, 187–196 (2016)
24. Zhang, X., Yin, L., Cohn, J.F., Canavan, S., Reale, M., Horowitz, A., Liu, P., Girard, J.M.: Bp4d-spontaneous: a high-resolution spontaneous 3d dynamic facial expression database. *Image and Vision Computing* **32**(10), 692–706 (2014)
25. Zhao, K., Chu, W.S., De la Torre, F., Cohn, J.F., Zhang, H.: Joint patch and multi-label learning for facial action unit detection. In: IEEE CVPR. pp. 2207–2216 (2015)
26. Zhao, K., Chu, W.S., Zhang, H.: Deep region and multi-label learning for facial action unit detection. In: IEEE CVPR. pp. 3391–3399 (2016)
27. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.H.: Conditional random fields as recurrent neural networks. In: IEEE CVPR. pp. 1529–1537 (2015)
28. Zhong, L., Liu, Q., Yang, P., Huang, J., Metaxas, D.N.: Learning multiscale active facial patches for expression analysis. *IEEE Transactions on Cybernetics* **45**(8), 1499–1510 (2015)