Image and Vision Computing 81 (2019) 1–14

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Learning facial action units with spatiotemporal cues and multi-label sampling☆

Wen-Sheng Chu[a,*], Fernando De la Torre[a], Jeffrey F. Cohn[a,b]

[a] Robotics Institute, Carnegie Mellon University, Pittsburgh, USA
[b] Department of Psychology, University of Pittsburgh, Pittsburgh, USA

ARTICLE INFO

Article history:
Received 16 October 2017
Received in revised form 17 May 2018
Accepted 22 October 2018
Available online 28 October 2018

Keywords:
Multi-label learning
Deep learning
Spatio-temporal learning
Multi-label sampling
Facial action unit detection
Video analysis

MSC:
00-01
99-00

ABSTRACT

Facial action units (AUs) can be represented *spatially*, *temporally*, and in terms of their *correlation*. Previous research focuses on one or another of these aspects or addresses them disjointly. We propose a hybrid network architecture that jointly models spatial and temporal representations and their correlation. In particular, we use a Convolutional Neural Network (CNN) to learn spatial representations, and a Long Short-Term Memory (LSTM) to model temporal dependencies among them. The outputs of CNNs and LSTMs are aggregated into a fusion network to produce per-frame prediction of multiple AUs. The hybrid network was compared to previous state-of-the-art approaches in two large FACS-coded video databases, GFT and BP4D, with over 400,000 AU-coded frames of spontaneous facial behavior in varied social contexts. Relative to standard multi-label CNN and feature-based state-of-the-art approaches, the hybrid system reduced person-specific biases and obtained increased accuracy for AU detection. To address class imbalance within and between batches during network training, we introduce multi-labeling sampling strategies that further increase accuracy when AUs are relatively sparse. Finally, we provide visualization of the learned AU models, which, to the best of our best knowledge, reveal for the first time how machines see AUs.

1. Introduction

Facial actions convey information about a person's emotion, intention, and physical state, and are vital for use in studying human cognition and related processes. To encode such facial actions, the Facial Action Coding System (FACS) [1,2] is the most comprehensive. FACS segments visual effects of facial activities into anatomically-based action units (AUs), which individually or in combinations can describe nearly all-possible facial expressions. Action unit description has led to multiple discoveries in behavioral and clinical science and other fields [2,3].

A conventional pipeline for automated facial AU detection compiles four stages: face detection ↦ alignment ↦ representation ↦ classification. With the progress made in face detection and alignment, research focuses on features, classifiers, or their combinations. At least three aspects affect the performance of automated AU detection: (1) *Spatial representation:* Hand-crafted features (*e.g.*, SIFT and HOG) have been widely used for AU detection, yet are susceptible to person-specific biases (*e.g.*, [4-6]). To be successful, representations must generalize to unseen subjects, regardless of individual differences caused by behavior, facial morphology and recording environments. (2) *Temporal modeling:* Action units are dynamic events. For this reason, temporal cues are critical to precise detection. An open research question is how to model dynamics and temporal context. (3) *AU correlation:* Action units are inter-dependent. Some actions are mutually exclusive (*e.g.*, open mouth cannot co-occur with closed mouth) while may increase or decrease the probability of other action units. For instance, AU12 (lip-corner pull) increases the likelihood of AU6 (contraction of the sphincter muscle around the eyes) and reduces the likelihood of AU15 (lip-corner depressor). By combining spatial representation, temporal modeling, and correlations among AUs, optimal detection performance can be achieved.

More specifically, we propose a hybrid network that jointly models spatial and temporal cues and the correlation among AUs. Fig. 1 gives an overview of the proposed framework. To learn a generalizable representation, a CNN is trained to learn and extract spatial features. To capture temporal dependencies, LSTMs are stacked on top of the spatial features. Lastly, we aggregate the learned representations from both CNNs and LSTMs into a fusion network that

☆ This paper has been recommended for acceptance by Yan Tong.
* Corresponding author.
E-mail address: wschu@cmu.edu (W.-S. Chu).

https://doi.org/10.1016/j.imavis.2018.10.002
0262-8856/

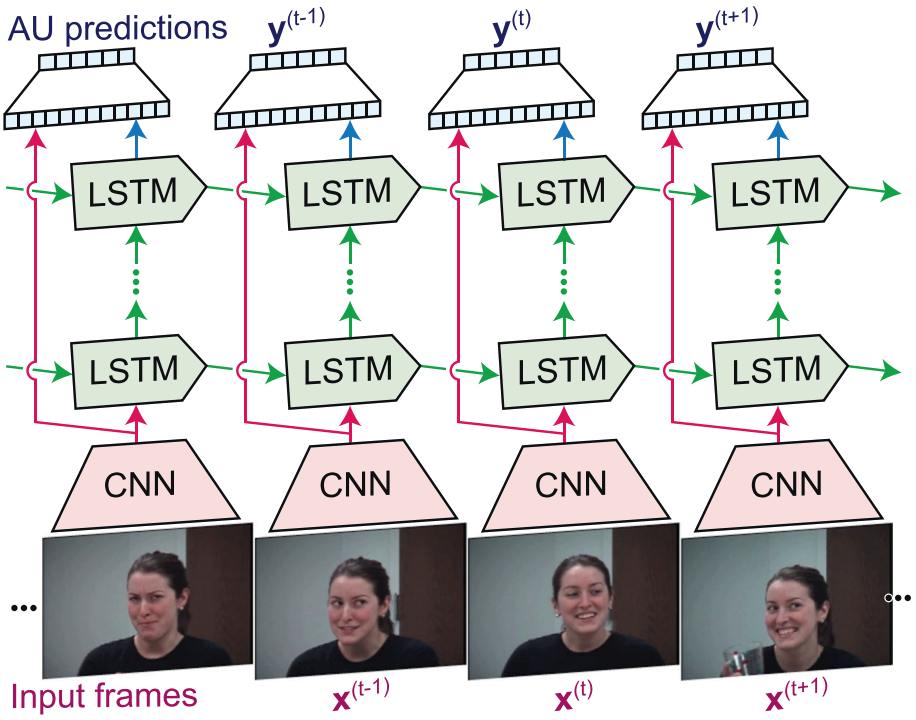


Fig. 1. An overview of the proposed hybrid network architecture: The proposed network possesses both strengths of CNNs and LSTMs to model both spatial and temporal cues, and combines both cues by a fusion network to produce frame-based prediction of multiple AUs.

predicts action units for each frame. Extensive experiments were performed on two spontaneous AU datasets, GFT and BP4D, containing totally >400,000 frames. The learned spatial features combined with temporal information outperformed a standard CNN and feature-based state-of-the-art methods. Quantitative and quantitative comparisons inform the advantages of the hybrid architecture relative to comparison approaches.

An earlier version of this paper appeared as [7]. In this paper, we introduce new multi-label sampling strategies and larger experiments to demonstrate that reducing class imbalance within and between batches during training further improves AU detection for AUs that have low base rates. The current paper is organized as follows. Section 3 presents the proposed hybrid network. Section 4 evaluates both the learned representation and the performance of the proposed network against alternative methods. Section 5 introduces multi-label sampling strategies and comparisons with conventional sampling approaches. Section 6 presents visualizations of the learned AU models. Section 7 concludes our findings and provides pointers to future work.

2. Related work

Below we review contemporary issues in automated facial AU detection and success in deep networks.

Facial AU detection: Despite advances in features, classifiers, and their combinations [8-11], three important aspects reside in automated AU detection. The first aspect is *spatial representation*, which is typically biased to individual differences such as appearance, behavior or recording environments. These differences produce shifted distributions in feature space (*i.e.*, *covariate shift*), hindering the generalizability of pre-trained classifiers. To reduce distribution mismatch, several studies merged into *personalization* techniques. Chu et al. [4] personalized a generic classifier by iteratively re-weighting training samples based on relevance to a test subject. Along this line, Sangineto et al. [5] directly transferred classifier parameters from source subjects to a test one. Zeng et al. [12] adopted an easy-to-hard strategy by propagating confident predictions to uncertain ones. Yang et al. [6] further extended personalization for estimating AU intensities by removing a person's identity with a latent factor model. Rudovic et al. [13] interpreted the person-specific variability as a context-modeling problem, and propose a conditional ordinal random field to address context effects. Others sought to learn AU-specific facial patches to specialize the representation [14,15]. However, while progress has been made, these studies still resort to hand-crafted features. We argue that person-specific biases from such features can be instead reduced by learning them.

Another aspect remains in *temporal modeling*, as modeling dynamics is crucial in human-like action recognition. To explore temporal context, graphical models have been popularly used for AU detection. A hidden CRF [16] classified over a sequence and established connections between the hidden states and AUs. These models made Markov assumption and thus lacked consideration of long-term dependencies. As an alternative, switching Gaussian process models [17] was built upon dynamic systems and Gaussian process to simultaneously track motions and recognize events. However, the Gaussian assumption unnecessarily holds in real-world scenarios. In this paper, we attempt to learn long-term dependencies to improve predicting AUs without the requirement to *a priori* of state dependencies and distributions.

Last but not the least, it has attracted an increasing attention on how to effectively incorporate *AU correlations*. Due to the fact that AUs could co-occur simultaneously within a frame, AU detection by nature is a *multi-label* instead of a *multi-class* classification problem as in holistic expression recognition, *e.g.*, [18,19]. To capture AU correlations, a generative dynamic Bayesian network (DBN) [20] was proposed with consideration of their temporal evolutions. Rather than learning, pairwise AU relations can be statistically inferred using annotations, and then injected into a multi-task framework to select important patches per AU [14]. In addition, a restricted Boltzmann machine (RBM) [21] was developed to directly capture

the dependencies between image features and AU relationships. Following this direction, image features and AU outputs were fused in a continuous latent space using a conditional latent variable model [22]. For the scenario with missing labels, a multi-label framework can be applied by enforcing the consistency between the predicted labels and the annotation [23]. Although improvements can be observed from predicting multiple AUs jointly, these approaches rely on engineered features such as HOG, LBP, or Gabor.

Deep networks: Recent success of deep networks suggests strategically composing nonlinear functions results in powerful models for perceptual problems. Closest to our work are the ones in AU detection and video classification.

Most deep networks for AU detection directly adapt CNNs (*e.g.*, [25]). Gadi et al. [26] used a 7-layer CNN for estimating AU occurrence and intensity. Ghosh et al. [27] showed that a shared representation can be directly learned from input images using a multi-label CNN. To incorporate temporal modeling, Jaiswal et al. [28] trained CNNs and BLSTM on shape and landmark features to predict for individual AUs. Because input features were predefined masks and image regions, unlike this study, gradient cannot backprop to full face region to analyze per-pixel contributions to each AU. In addition, it ignored AU dependencies and temporal info that could improve performance in video prediction, *e.g.*, [29,30]. On the contrary, our network simultaneously models spatial-temporal context and AU dependencies, and thus serves as a more natural framework for AU detection.

The construction of our network is inspired by recent studies in video classification. Simonyan et al. [29] proposed a two-stream CNN that considers both static frames and motion optical flow between frames. A video class was predicted by fusing scores from both networks using either average pooling or an additional SVM. To incorporate "temporally deep" models, Donahue et al. [31] proposed a general recurrent convolutional network that combines both CNNs and LSTMs, which can be then specialized into tasks such as activity recognition, image description and video description. Similarly, Wu et al. [30] used both static frames and motion optical flow, combined with two CNNs and LSTMs, to perform video classification. Video-level features and LSTM outputs were fused to produce a per-video prediction.

Our approach fundamentally differs from the above methods in several aspects: (1) Video classification is a *multi-class* classification problem, yet AU detection is *multi-label*. (2) Motion optical flow is usually useful in video classification, but *not* in AU detection due to large head movements. (3) AU detection requires *per-frame* detection; video classification produces *video-based* prediction.

3. The hybrid network for multi-label facial AU detection

Fig. 2(a) shows a folded illustration of the proposed hybrid network. Below we describe each component in turn.

3.1. Learning spatial cues with CNN

The literature has shown evidence that hand-crafted features impair generalization of AU detectors [4-6]. We argue that specialized representation could be learned to reduce the burden of designing sophisticated models, and further improve performance. On the other hand, some AUs co-occur frequently (*e.g.*, AUs 6+12 in a Duchenne smile), and some infrequently. Classifiers trained with AU relations were shown to lead to more reliable results [14,22,23]. To these two ends, we train a multi-label CNN by modifying the AlexNet [24] as shown in Fig. 2(b). Given a ground truth label $\mathbf{y} \in \{-1, 0, +1\}^L$ ($-1/+1$ indicates absence/presence of an AU, and 0 missing label) and a prediction vector $\widehat{\mathbf{y}} \in \mathbb{R}^L$ for L AU labels, this multi-label CNN aims to minimize the multi-label cross entropy loss:

$$L_E(\mathbf{y}, \widehat{\mathbf{y}}) = \frac{-1}{L} \sum_{\ell=1}^{L} [y_\ell > 0] \log \widehat{y}_\ell + [y_\ell < 0] \log(1 - \widehat{y}_\ell), \tag{1}$$

where $[x]$ is an indicator function returning 1 if x is true, and 0 otherwise. The outcome of the fc7 layer is L_2-normalized as the

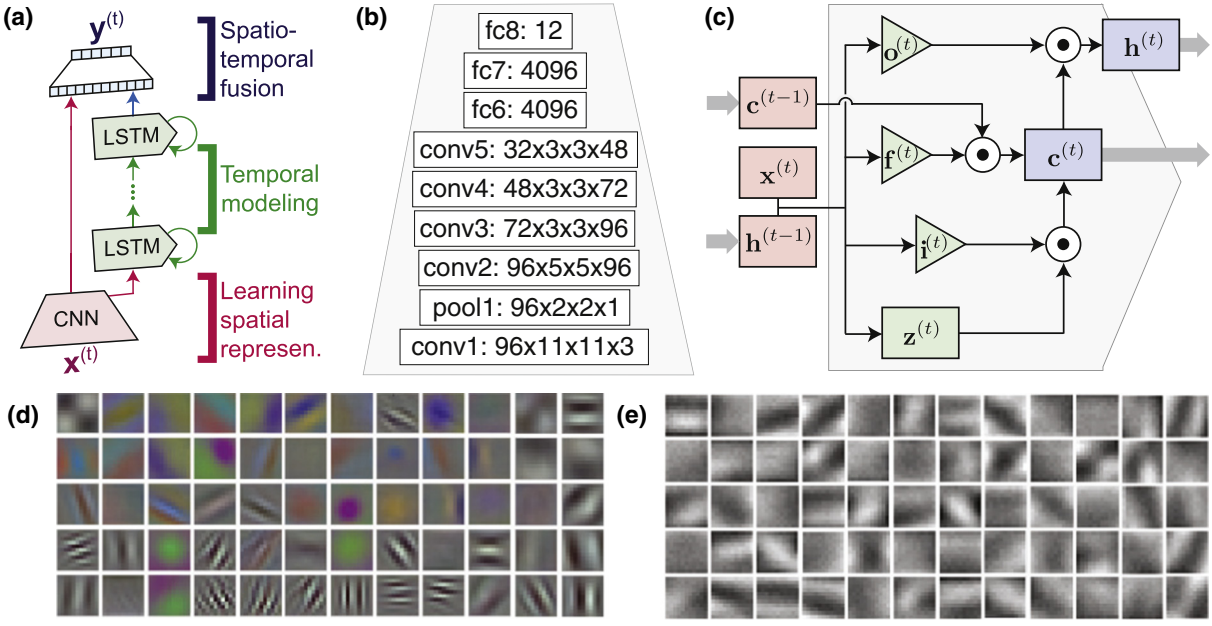


Fig. 2. The structure of the proposed hybrid network: (a) Folded illustration of Fig. 1, showing 3 components of *learning spatially representation*, *temporal modeling*, and *spatiotemporal fusion*, (b) our 8-layer CNN architecture, and (c) the schematic of an LSTM block. (d)–(e) Visualization of conv1 layers of models trained on ImageNet [24] and GFT datasets, respectively. As can be seen, filters learned on our face dataset contain less color blob detectors, suggesting color is less informative in AU detection.

final representation, resulting in a 4096-D vector. We denote this representation as "fc7" hereafter.

Note that we do not explicitly impose AU relations during learning (*e.g.*, add extra constraints). Instead, because multiple labels are assigned to each instance (due to the nature of multi-label architecture), the relationship among AUs is implicitly coded during training. For instance, when a smile face is present, the network is guided to predict AU 6+12 without knowing their relation. This is confirmed by the visualization of AU models as will be discussed in Section 6.

Fig. 2(d) and (e) visualizes the learned kernels from the conv1 layer on the ImageNet [24] and the GFT datasets, respectively. As can be seen, the kernels learned on GFT contain less color blob detectors than the ones learned on ImageNet. This suggests that color info is less useful in faces than in natural images. Similar patterns were observed on the BP4D dataset [32]. In Section 4, we will empirically evaluate fc7 against hand-crafted features such as the popular HOG and Gabor.

3.2. Learning temporal cues with stacked LSTMs

It is usually hard to tell an "action" by observing only a single frame. Having fc7 extracted, we used stacked LSTMs [33] for encoding temporal context. Fig. 2(c) shows the schematic of a standard LSTM block. Unlike learning spatial representation on cropped face images, videos can be difficult to model with a fixed-size architecture, *e.g.*, [16,34]. LSTM serves as an ideal candidate for learning long-term dependencies, and avoids the well-known "vanishing gradient" in recurrent models. Due to an absence of theory in choosing the number of LSTM layers and size of each memory cell, we took an empirical approach by considering the tradeoff between accuracy and computational cost, and ended up with 3 stacks of LSTMs with 256 memory cells each.

AU detection is by nature a *multi-label classification* problem. We optimize LSTMs to jointly predict multiple AUs according to the maximal-margin loss: $L_M(\mathbf{y},\widehat{\mathbf{y}}) = \frac{1}{n_0}\sum_i \max(0,\lambda - y_i\widehat{y}_i)$, where λ is a pre-defined margin, and n_0 indicates the number of non-zero elements in ground truth $\mathbf{y}$. One reason for using max-margin instead of cross-entropy loss as in Section 3.1 is due to more uncertainties in temporal modeling than in spatial modeling. For instance, head motions and different duration and speed in actions could cause the temporal patterns of AUs less structured and thus harder to learn the per-frame prediction. Max-margin loss could potentially permit some tolerance (*i.e.*, samples beyond the margin were ignored) instead of enforcing LSTMs to match every frame. Although typically $\lambda = 1$ (such as in regular SVMs), here we empirically choose $\lambda = 0.5$ because the activation function has squeezed the outputs into $[-1,1]$, making the prediction value never go beyond $\lambda = 1$. During back propagation, we pass the gradient $\frac{\partial L}{\partial \widehat{y}_i} = -\frac{y_i}{n_0}$ if $y_i\widehat{y}_i < 1$, and $\frac{\partial L}{\partial \widehat{y}_i} = 0$ otherwise. At each time step, LSTMs output a vector indicating potential AUs.

Practical issues: There has been evidence that a deep LSTM structure preserves better descriptive power than a single-layer LSTM [33]. However, because fc7 features are of high-dimension (4096-D), our design of LSTMs can lead to a large model with >1.3 million parameters. To ensure that the number of parameters and the size of our datasets maintain the same order of magnitude, we applied PCA to reduce the fc7 features to 1024-D (preserving 98% energy). We set dropout rate as 0.5 to the input and hidden layers, resulting in a final model of ~0.2 million parameters. More implementation details are in Section 4.

3.3. Fusing spatial and temporal cues

The spatial CNN performs AU detection from still video frames, while the temporal LSTM is trained to detect AUs from temporal transitions. Unlike video classification that produces video-based prediction (*e.g.*, [29-31]), we model the correlations between spatial and temporal cues by adding an additional fusion network. We modify the late fusion model [34] to achieve this goal. Fig. 1 shows an illustration. For each frame, two fully connected layers with shared parameters are placed on top of both CNNs and LSTMs. The fusion network merges the stacked L_2-normalized scores in the first fully connected layer. In experiments, we see that this fusion approach consistently improves the performance compared to CNN-only results.

4. Experiments

4.1. Datasets

We evaluated the proposed hybrid network on two large spontaneous datasets: BP4D [32] and GFT [35] consisting of >400,000 manually FACS-coded frames. AUs occurring more than 5% base rate were included for analysis. Twelve AUs met this criterion. Unlike the previous studies that suffer from scalability issues and require downsampling of training data, the network is in favor of large dataset so we made use of all available data.

BP4D [32] is a spontaneous facial expression dataset in both 2D and 3D videos. The dataset includes 328 videos of approximately 20 s each from a total of 41 participants. Action unit intensity was available for the full range from A through E. We selected positive samples as those with intensities equal or higher than A-level, and negative samples as the remaining.

GFT [35] contains 240 groups of three previously unacquainted young adults. Moderate out-of-plane head motion and occlusion are presented in the videos. We used 50 participants with each containing one video of about 2 min (~5000 frames), resulting in 254,451 available frames. Action unit intensity was available for B through E intensity. A, or trace level, was not coded. Frames with intensities equal or greater than B-level were used as positive; other frames were regarded negative.

4.2. Settings

Pre-processing: We pre-processed all videos by extracting facial landmarks using IntraFace [36]. Tracked faces were registered to a reference face using similarity transform, resulting in 200×200 face images, which were then randomly cropped into 176×176 and/or flipped for data augmentation. Each frame was labeled +1/−1 if an AU is present/absent, and 0 otherwise (*e.g.*, lost face tracks or occluded face).

Dataset splits: For both datasets, we adopted a *3-fold* and a *10-fold* protocol to evaluate the effect of the number of training samples and the generalization capability of different methods, *i.e.*, the 10-fold protocol uses ~30% more samples than the 3-fold. For 3-fold protocol, each dataset was evenly divided into 3 subject-exclusive partitions. We iteratively trained a model using two partitions and evaluated on the remaining one, until all subjects were tested. For 10-fold protocol, we followed standard train/validation/test splits as in the deep learning community (*e.g.*, [24,29,30]). In specific, we divided entire dataset into 10 subject-exclusive partitions, where 9 for training/validation and 1 for test. For both protocols, we used ~20% training subjects for validation. To measure the transferability of fc7 features, we also performed a *cross-dataset* protocol by training CNNs on one dataset and using it to extract spatial representations for training a classifier on another.

Evaluation metrics: We reported performance using frame-based F1-score (F1-frame$=\frac{2RP}{R+P}$) for comparisons with the literature, where R and P denote recall and precision, respectively. In addition, because AUs occur as temporal signals, an event-based F1 (F1-event$=\frac{2ER\bullet EP}{(ER+EP)}$) can be used to measure detection performance

at segment-level, where *ER* and *EP* are event-based recall and precision as defined in [37]. Different metrics capture different properties about the detection performance. Choices of one or another metric depend on a variety of factors, such as purposes of the task, preferences of individual investigators, the nature of the data. Due to space limitation, we only reported F1 in this paper.

Network settings and training: We trained the CNNs with mini-batches of 196 samples, a momentum of 0.9 and weight decay of 0.0005. All models were initialized with learning rate of 1e-3, which was further reduced manually whenever the validation loss stopped decreasing. The implementation was based on the Caffe toolbox [38] with modifications to support multi-label cross-entropy loss. For training LSTMs, we set an initial learning rate of 1e-3, momentum of 0.9, weight decay 0.97, and RMSProp for stochastic gradient descent. All gradients were computed using back-propagation through time (BPTT) on 10 subsequences randomly sampled from training video. All sequences were 1300-frame long, and the first 10 frames were disregarded during the backward pass, as they carried insufficient temporal context. The matrix $\mathbf{W}$ were randomly initialized within $[-0.08, 0.08]$. As AU data is heavily skewed, randomly sampling the sequences could cause LSTMs biased to negative predictions. As a result, we omitted training sequences with less than 1.5 active AUs per frame. All experiments were performed using one NVidia Tesla K40c GPU.

4.3. Evaluation of learned representation

To answer the question whether individual differences can be reduced by feature learning, we first evaluated the fc7 features with standard features in AU detection, including shape (landmark locations), Gabor, and HOG features. Because such features for AU detection are unsupervised, for fairness, fc7 features for BP4D were extracted using CNNs trained on GFT, and vise versa. Fig. 3 shows the t-SNE embeddings of frames represented by HOG, VGG face descriptor [39] and fc7 features colored in terms of AU12 and subject identities. As can be seen in the first and second columns, HOG and VGG face descriptors have strong distributional biases toward subject identity. On the other hand, as shown in the third column, although the network is learned on the other dataset, fc7 features show relative invariance to individual differences. More importantly, as shown in the plot of AU12, fc7 features maintain the grouping effect on samples of the same AU, implying its ability of capturing necessary information for classification.

As a quantitative evaluation, we treated the frames of the same subject as a distribution, and computed the distance between two subjects using Jensen-Shannon (JS) divergence [40]. Explicitly, we first computed a mean vector $\boldsymbol{\mu}_s$ for each subject s in the feature space, and then squeezed $\boldsymbol{\mu}_s$ using a logistic function $\sigma(a) = \frac{1}{1+e^{-a/m}}$ (m is median of $\boldsymbol{\mu}_s$ as the median heuristic) and unity normalization, so that each mean vector can be interpreted as a discrete probability distribution, *i.e.*, $\boldsymbol{\mu}_s \geq 0$, $\|\boldsymbol{\mu}_s\|_1 = 1, \forall s$. Given two subjects p and q, we computed their JS divergence as:

$$D(\boldsymbol{\mu}_p, \boldsymbol{\mu}_q) = \frac{1}{2} D_{\mathrm{KL}}(\boldsymbol{\mu}_p || \mathbf{m}) + \frac{1}{2} D_{\mathrm{KL}}(\boldsymbol{\mu}_q || \mathbf{m}), \tag{2}$$

where $\mathbf{m} = \frac{1}{2}(\boldsymbol{\mu}_p + \boldsymbol{\mu}_q)$ and $D_{\mathrm{KL}}(\boldsymbol{\mu}_p, \mathbf{m})$ is the discrete KL divergence of $\boldsymbol{\mu}_p$ from $\mathbf{m}$. JS divergence is symmetric and smooth, and has been

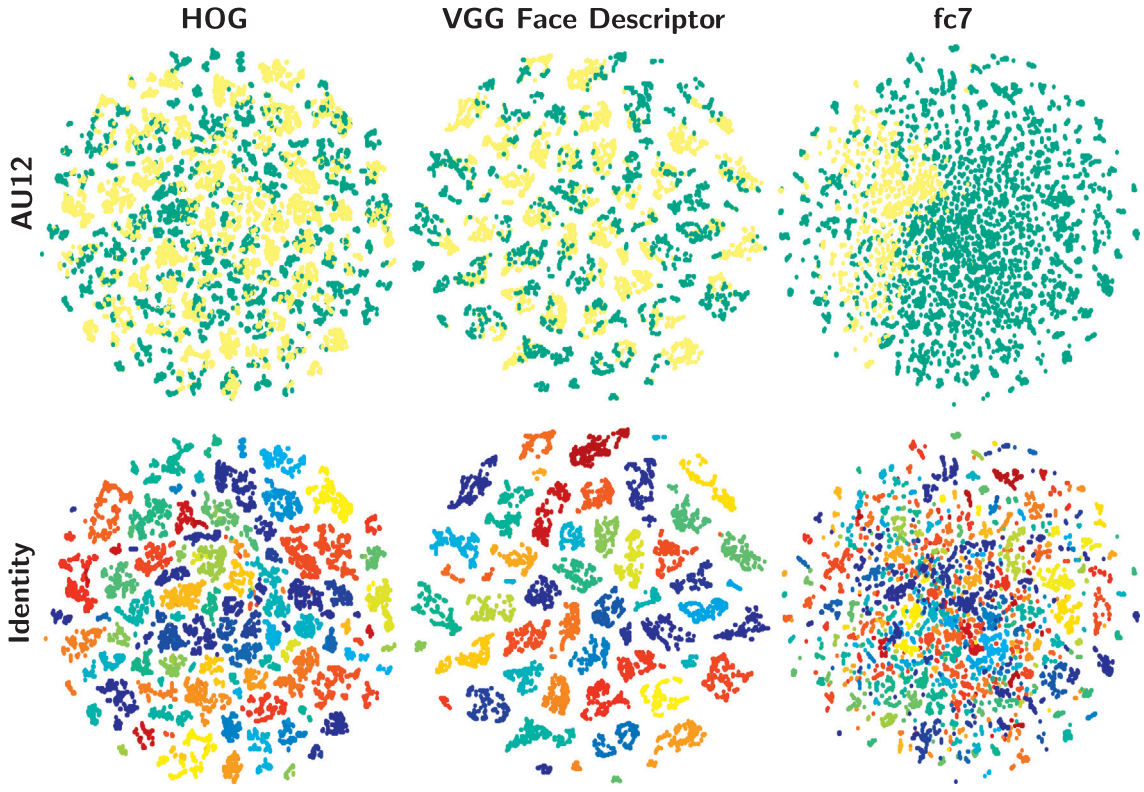


Fig. 3. A visualization of t-SNE embedding using HOG, VGG face descriptor [39] and fc7 features on the BP4D dataset [32] by coloring each frame sample in terms of AU12 (top row) and subject identities (bottom row). The clustering effect in HOG and VGG face descriptors reveals their encoded information about not only facial AUs but more subject identities. As can be seen, the separation between subjects of VGG descriptors is more clear than the separation of HOG, because VGG face descriptors were originally trained for face recognition. On the other hand, the learned fc7 features are optimized for multi-label AU classification, and thus reduce the influence caused by individual differences. More importantly, fc7 features maintain the grouping effect on samples of the same AU, implying its ability of capturing necessary information for AU classification.

Dataset	Shape	Gabor	HOG	fc7
BP4D	5.38±.40	4.63±.16	3.87±.12	3.58±.09
GFT	5.43±.39	4.74±.23	3.41±.25	0.89±.13

Fig. 4. Subject-invariance on the BP4D and GFT datasets in terms of a computed JS-divergence d normalized by $\log(d)\times 1e6$ (details in text).

shown effective in measuring the dissimilarity between two distributions (*e.g.*, [41]). Higher value of $D(\boldsymbol{\mu}_p, \boldsymbol{\mu}_q)$ tells larger mismatch given distributions for two subjects. Fig. 4 shows the statistics of distributional divergence over all subjects in one dataset, which was computed by summing over $D(\boldsymbol{\mu}_p, \boldsymbol{\mu}_q), \forall q \neq p$. As can be seen, HOG consistently reached a lower divergence than Gabor, providing an evidence that local descriptor (HOG) is more robust to appearance changes compared to holistic ones (Gabor). This also serves as a possible explanation why HOG consistently outperformed Gabor (*e.g.*, [42]). Overall, fc7 yields much lower divergence compared to alternative engineered features, implying reduced individual differences.

4.4. Evaluation of detection performance

This section evaluates the detection performance of the proposed network on BP4D and GFT datasets. Below we summarize alternative methods, and then provide results and discussion.

Alternative methods: We compared a baseline SVM trained with HOG features, a standard multi-label CNN, and feature-based state-of-the-arts. The HOG features have been shown to outperform other appearance descriptors (*e.g.*, Gabor/Daisy) [42]. Because HOG is unsupervised, for fairness, we evaluated a *cross-dataset* protocol that trained an AlexNet on the other dataset, termed as ANet$^{\mathsf{T}}$. fc7 features extracted by ANet$^{\mathsf{T}}$ were then used in comparison with HOG descriptors. Linear SVMs served as the base classifier, which implicitly tells how separable each feature was, *i.e.*, higher classification rate suggests an easier linear separation, and validates that a good representation could reduce the burden of designing a sophisticated classifier. We evaluated ANet$^{\mathsf{T}}$ on a 3-fold protocol, while we expect that similar results could be obtained using 10-fold.

Another alternative is our modified AlexNet (ANet), as mentioned in Section 3.1, with slightly different architecture and loss function (multi-label cross-entropy instead of multi-class softmax). ANet stood for a standard multi-label CNN, a representative of *feature learning* methods. On the other hand, CPM [12] and JPML [14] are feature-based state-of-the-art methods that were reported on the two datasets. Both CPM and JPML used HOG features [12,14]. They differ in attacking the AU detection problem from different perspectives. CPM is one candidate method of *personalization*, which aims at identifying reliable training samples for adapting a classifier that best separates samples of a test subject. On the other hand, JPML models *AU correlations*, and meanwhile considers patch learning to select important facial patches for specific AUs. We ran all experiments following protocols in Section 4.2.

Results and discussion: Tables 1 and 2 show F1 metrics reported on 12 AUs; "Avg" for the mean score of all AUs. According to the results, we discuss our findings in hope to answer three fundamental questions:

1) *Could we learn a representation that better generalizes across subjects or datasets for AU detection?* On both datasets, compared to the baseline SVM, ANet$^{\mathsf{T}}$ trained with a cross-dataset protocol on average yielded higher scores with a few exceptions. In addition, for both 3-fold and 10-fold protocols where ANet was trained on exclusive subjects, ANet consistently outperformed SVM over all AUs. These observations provide an encouraging evidence that the learned representation was transferable even when being tested across subjects and datasets, which also coincides with the findings in the image and video classification community [29, 34, 43]. On the other hand, as can be seen, ANet trained within datasets leads to higher scores than ANet$^{\mathsf{T}}$ trained across datasets. This is because of the dataset biases (*e.g.*, recording environment, subject background) that could cause distributional shifts in the feature space. In addition, due to the complexity of deep models, the performance gain of ANet trained on more data (10-fold) became larger than ANet trained on 3-fold, showing that the generalizability of deep models increases with the growing number of training samples. Surprisingly, compared to SVM trained on 10-fold, ANet trained on 3-fold showed comparable scores, even with ∼30% fewer data than what SVM was used. All suggests that features less sensitive to the identity of subjects could improve AU detection performance.
2) *Could the learned temporal dependencies improve performance, and how?* The learned temporal dependencies was aggregated into the hybrid network denoted as "ours". On both 3-fold and 10-fold protocols, our hybrid network consistently outperformed ANet in all metrics. This improvement can be better told by comparing their F1-event scores. The proposed network used CNNs to extract spatial representations, stacked LSTMs to model temporal dependencies, and then performs a spatiotemporal fusion. From this view, predictions with fc7 features can be treated as a spacial case of ANet—a linear

Table 1
F1-frame on GFT dataset [35].

	3-Fold protocol					Cross	10-Fold protocol				
AU	SVM	CPM	JPML	ANet	Ours	ANet$^{\mathsf{T}}$	SVM	CPM	JPML	ANet	Ours
1	12.1	30.7	17.5	**31.2**	29.9	9.9	30.3	29.9	28.5	57.5	**63.0**
2	13.7	30.5	20.9	29.2	25.7	10.8	25.6	25.7	25.5	61.4	**74.6**
4	5.5	–	3.2	**71.9**	68.9	45.4	–	–	–	**75.9**	68.5
6	30.6	61.3	**70.5**	64.5	67.3	46.2	66.2	67.3	73.1	61.6	**66.3**
7	26.4	70.3	65.5	67.1	**72.5**	51.5	70.9	72.5	70.2	**80.1**	74.5
10	38.4	65.9	**67.9**	42.6	67.0	23.5	65.5	67.0	67.1	54.5	**70.3**
12	35.2	74.0	74.2	73.1	**75.1**	55.2	74.2	75.1	78.3	**79.8**	78.2
14	55.8	**81.1**	52.4	69.1	80.7	62.8	79.6	80.7	61.4	**84.2**	80.4
15	9.5	25.5	20.3	27.9	**43.5**	14.2	34.1	43.5	28.0	40.3	**50.5**
17	31.3	44.1	48.3	**50.4**	49.1	34.2	49.2	49.1	42.4	61.6	**61.9**
23	19.5	19.9	31.8	34.8	**35.0**	21.8	28.3	35.0	29.6	47.0	**58.2**
24	12.9	27.2	28.5	**39.0**	31.9	18.9	31.9	31.6	28.0	**56.3**	50.8
Avg	24.2	48.2	41.8	50.0	**53.9**	32.9	50.5	52.4	48.4	63.4	**66.4**

Table 2
F1-frame metrics on BP4D dataset [32].

	3-Fold protocol					Cross	10-Fold protocol				
AU	SVM	CPM	JPML	ANet	Ours	ANet^T	SVM	CPM	JPML	ANet	Ours
1	21.1	**43.4**	32.6	40.3	31.4	32.7	46.0	46.6	33.9	54.7	**70.3**
2	20.8	**40.7**	25.6	39.0	31.1	26.0	38.5	38.7	36.2	56.9	**65.2**
4	29.7	43.3	37.4	41.7	**71.4**	29.0	48.5	46.5	42.2	**83.4**	83.1
6	42.4	59.2	42.3	62.8	**63.3**	61.9	67.0	68.4	62.9	94.3	**94.7**
7	42.5	61.3	50.5	54.2	**77.1**	59.4	72.2	73.8	69.9	93.0	**93.2**
10	50.3	62.1	72.2	**75.1**	45.0	67.4	72.7	74.1	72.5	98.9	**99.0**
12	52.5	68.5	74.1	78.1	**82.6**	76.2	83.6	84.6	72.0	94.4	**96.5**
14	35.2	52.5	65.7	44.7	**72.9**	47.1	59.9	62.2	62.6	82.9	**86.8**
15	21.5	**36.7**	38.1	32.9	34.0	21.7	41.1	44.3	38.2	55.4	**63.3**
17	30.7	**54.3**	40.0	47.3	53.9	47.1	55.6	57.5	46.5	81.1	**82.7**
23	20.3	**39.5**	30.4	27.3	38.6	21.6	40.8	41.7	38.3	63.7	**73.5**
24	23.0	37.8	42.3	**40.1**	37.0	31.3	42.1	39.7	41.5	74.3	**81.6**
Avg	32.5	50.0	45.9	48.6	**53.2**	43.4	55.7	56.5	51.4	77.8	**82.5**

hyperplane with a portion of intermediate features. In general, adding temporal information helped predict AUs except for a few in GFT. A possible explanation is that in GFT, the head movement was more frequent and dramatic, and thus makes temporal modeling of AUs more difficult than moderate head movements in BP4D. In addition, adding temporal prediction into the fusion network attained an additional performance boost, leading to the highest F1 score on both datasets with either the 3-fold or the 10-fold protocols. Note that using solely the temporal model causes slight performance drop, as temporal transition could be unclear to capture in the spontaneous datasets (*e.g.*, mouth motions due to speech), and might require more complex models (*e.g.*, bi-directional LSTMs). We leave deeper investigation into more sophisticated temporal models in future work. In all, similar to "late fusion", the overall network shows that the spatial and temporal cues are complementary, and thus is crucial to incorporate all of them into an AU detection system.

3) *Would jointly considering all issues in one framework improve AU detection?* This question aims to examine if the hybrid network would improve the performance of the methods that consider the aforementioned issues independently. To answer this question, we implemented CPM [12] as a personalization method that deals with representation issues, and JPML [14] as a multi-label learning method that deals with AU relations. Our modified ANet served as a feature learning method. All parameters settings were determined following the descriptions in the original papers. To draw a valid discussion, we fixed the exact subjects for all methods. Observing 3-fold on both datasets, the results are mixed. In GFT, ANet and JPML achieved 3 and 2 highest F1 scores; in BP4D, CPM and ANet reached 5 and 2 highest F1 scores. An explanation is because, although CNNs possess high degree of expressive power, the number training samples in 3-fold (33% left out for testing) were insufficient and might result in overfitting. In the 10-fold experiment, when training data was abundant, the improvements became clearer, as the parameters of the complex model can better fit our task. Overall, in most AUs, our hybrid network outperformed alternative approaches by a significant margin, showing the benefits for considering all issues in one framework.

Note that the proposed approach does not always outperform the others due to generalizability issues of the model. This can attribute to multiple factors in data (*e.g.*, label consistency, insufficient diversity in subjects or samples, data bias) and algorithmic design (*e.g.*, network architecture, training strategy, model selection), which remain an open question for future investigation.

5. Multi-label sampling

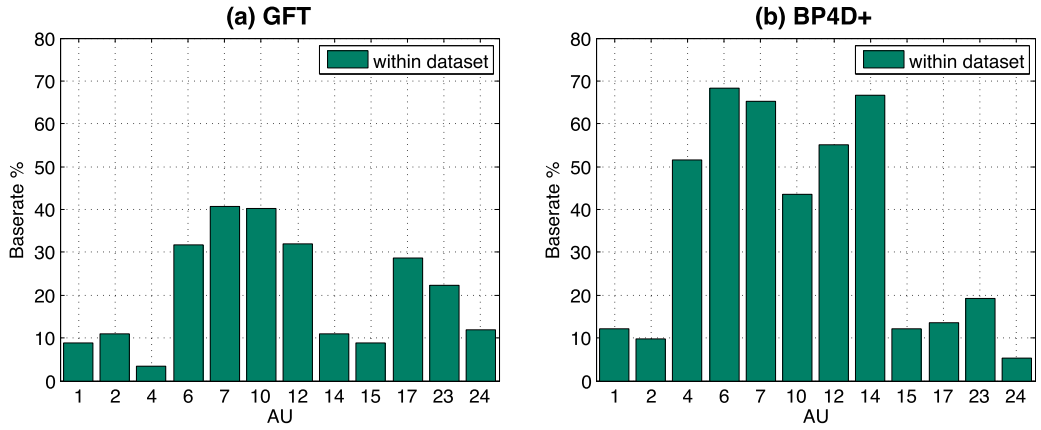
In spontaneous datasets, the incidence of AU labels can vary greatly. As shown in Fig. 5, certain AUs occur with high base rate (*e.g.*, AUs 6, 7 and 12) while others occur infrequently (*e.g.*, AUs 1, 2 and 15). Without any treatment on class imbalance, classifiers trained on this distribution could cause predictions biased by major classes (classes with higher base rate) due to a global error measurement. That is, when unbalanced class distribution is present, minor classes do not contribute equally in the global performance measure, resulting in a natural inclination to the most frequent classes. As can be observed in the Section 4.4, the hybrid multi-label network performed less well for minor classes, such as AUs 1, 2, and 15.

In general, performance on rare classes can be improved if more samples are observed. We refer interested readers to more comprehensive reviews (*e.g.*, [45-47]). In the literature, the imbalance levels are often referred to as *imbalance ratio* or *skewness*, computed as the ratio of the size of the majority class over the size of the minority class. Standard approaches in learning from such unbalanced classes can be broadly categorized into two branches:

- **Resampling:** Resampling techniques aim at producing a new dataset from the original one. To balance the distributions between frequently and rarely occurring classes, oversampling or undersampling approaches are typically used. Another trend employs synthesis for the minority class, *i.e.*, growing the population of minority classes by synthesizing samples in the feature space (*e.g.*, SMOTE [48]). Because the sampling is done at data-level, resampling can be seen as a classifier-independent approach that applies to most problems.
- **Classifier adaptation/cost-sensitive learning:** This type of methods is classifier-dependent. The goal here is to modify a classification algorithm to further emphasize the contributions of minor classes. The unbalanced nature of the data is addressed by re-estimating sample distribution, reinforcing the algorithm toward the minority class, or re-weighting training losses inversely proportional to each class size.

Although imbalance learning has been a well-known problem with rather comprehensive studies, most existing methods only consider sampling for only one majority class and one minority class. Because facial images contain several AU class labels per sample, the complexity of the sampling problem becomes higher, making the application of standard approaches indirectly applicable.

As illustrated in Fig. 5, a clear imbalance among AU classes exists in spontaneous datasets, such as GFT [49] and BP4D+ [44]. For instance, in BP4D+, the most frequently occurring AU has



(c) Base rate (%)

Dataset	1	2	4	6	7	10	12	14	15	17	23	24
GFT	12.1	9.7	51.5	68.2	65.3	43.5	55.1	66.8	12.1	13.6	19.2	5.3
BP4D+	8.8	11.0	3.4	31.8	40.7	40.3	32.0	10.9	8.8	28.7	22.2	12.0

Fig. 5. Distributions of AU base rates in two of the largest spontaneous datasets used in this study: (a) GFT [35] and (b) BP4D+ [44]. (c) Shows the exact base rate of individual AUs of each dataset. Base rate is defined as the frequency of a particular AU occurring in video frames of the entire dataset. Note that we only count the frames that can be validly face tracked and annotated completely with 12 AUs.

more than 10 times more samples than the least occurring one. We note that, for illustrating the severity of class imbalance, we used larger, renewed GFT and BP4D+ datasets in this section than earlier experiments in Section 4.4. Recall that in an end-to-end supervised framework, "batches" are randomly sampled from the training set for updating parameters in stochastic gradient descent. However, randomly selecting images causes at least two issues for proper stochastic training. First, as illustrated in the top row of Fig. 6, the number of AU presence *between* batches is unbalanced. This can potentially weaken gradient stability between batches during back propagation. Second, the number of AU presence *within* batches is also unbalanced. As noted earlier, such imbalance can cause the learned model to favor the majority class. Due to these differences between AU class distributions, a multi-label sampling strategy is of specific need.

In this section, we will introduce two multi-label sampling strategies to attack this specific imbalance in the multi-label space: multi-label stratification in Section 5.1, and multi-label minority oversampling majority undersampling (MOMU) in Section 5.2. Then, in Section 5.3, we will evaluate different multi-label sampling strategies in both training and test phases.

Algorithm 1. Multi-label stratification.

Input : Dataset $\mathcal{D}$ annotated with L classes, the number of batches B
Output: Processed batches $\mathcal{B} = \{\mathcal{B}_i\}_{i=1}^{B}$

Compute N_ℓ ($\ell = 1, ..., L$) as the number of the ℓ-th AU in the dataset $\mathcal{D}$;
while $|\mathcal{D}| > 0$ **do**
  $\ell \leftarrow \arg\min_j N_j$; // Find the AU with fewest samples in $\mathcal{D}$
  $\mathcal{D}_\ell \leftarrow \{\{(\mathbf{x}_i, Y_i)\}_i \in \mathcal{D} | Y_i^\ell = 1\}$ // Collect (image,label) of the AU with fewest samples
  if *any of* $\mathcal{B}_i$ *is not full* **then**
    Distribute $\mathcal{D}_\ell$ evenly into all batches $\{\mathcal{B}_i\}_{i=1}^{B}$;
  end
  $\mathcal{D} \leftarrow \mathcal{D} \setminus \mathcal{D}_\ell$;
  Update N_ℓ ($\ell = 1, ..., L$) as the number of the ℓ-th AU;
end
return $\mathcal{B}$;

5.1. Multi-label stratification

We first propose an algorithm for balancing the distribution *between* batches. The idea was inspired by standard methods on *stratified sampling*, which utilizes independent sampling among each sub-population when sub-populations vary within an overall population. Algorithm 1 summarizes the proposed multi-label stratification approach. The input to the algorithm is a dataset $\mathcal{D} = \{\mathbf{x}_i, Y_i\}_{i=1}^{|\mathcal{D}|}$ annotated with L classes (*i.e.*, $Y_i \in \mathbb{R}^L$). Suppose $|\mathcal{D}|$ is the number of images in the dataset, and Y_i^ℓ is the ℓ-th AU annotation of the i-th image. The multi-label stratification starts by computing the total number of examples for each AU class, and then iteratively distributing images that contain the AU with the fewest samples. The distribution is performed evenly into each batch until the complete dataset is distributed ($|\mathcal{D}| = 0$) or the desired number of batches is collected. This normally terminates after ($L + 1$) iterations (L iterations for distributing all AUs and 1 iteration for distributing samples with no AUs annotations), but could end up less if samples of certain AU class have been already distributed. Note that images without any AU annotations still carry information about being an opposite (negative) class for each AU, and thus we enforce the sampling to terminate until the dataset is empty.

This algorithm is performed in a greedy perspective. That is, we aim to have labels in every batch as diverse as possible. If images that contain minority class labels are not evenly distributed in priority, it is likely that some batches contain zero occurrence of rare labels, resulting in biased learning that is difficult to be repaired subsequently. On the other hand, due to the availability of more samples, distributing later the images with labels from the majority classes maintains to guide the model towards a desired parametric update.

The middle row of Fig. 6 illustrates the distribution of AU presence in each mini-batch. As can be seen, the number of AU presence is much more balanced between batches compared to the random sampling shown in the first row. However, each vertical slice (*i.e.*, AU distribution in one batch) still exhibits dramatic unbalanced AU distribution. For example, minor AUs (*e.g.*, 1, 2 and 4) are outnumbered by major AUs (*e.g.*, 7, 10, and 12). To balance the distribution of AU presence within batches, we are driven to the next sampling strategy.

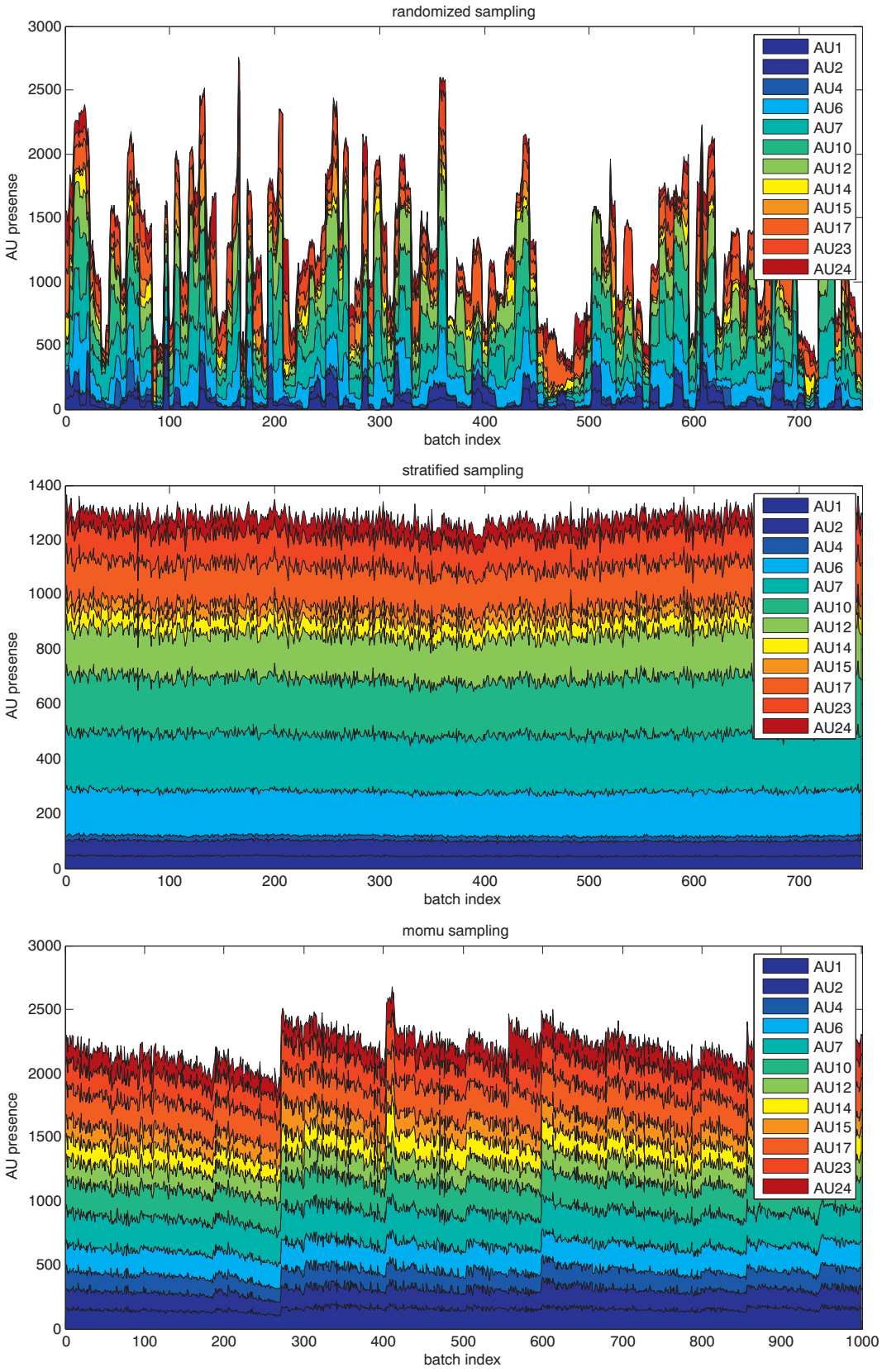


Fig. 6. Distributions of AU classes in each mini-batch using different sampling strategies: (top) random sampling, (middle) multi-label stratification, (bottom) MOMU sampling. As can be seen in random sampling, the number of AU presence *between* and *within* batches are dramatically different (see text for details).

Algorithm 2. The proposed multi-label minority oversampling majority undersampling (MOMU).

Input : Dataset $\mathcal{D}$ annotated with L labels, the size of a mini-batch N, the number of batches B, sampling size S
Output: Processed batches $\mathcal{B} = \{\mathcal{B}_i\}_{i=1}^{B}$

Compute N_ℓ ($\ell = 1, ..., L$) as the number of the ℓ-th AU in the dataset $\mathcal{D}$;
for $i = 1, ..., B$ **do**
  $\ell \leftarrow \arg\min_j N_j$;
  $\mathcal{B}_i \leftarrow \emptyset$; // Initialize current batch
  while $|\mathcal{B}_i| < N$ **do**
    if $N_\ell < S$ **then**
      $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{x}_i, Y_i)\} \in \mathcal{B} | Y_i^\ell = 1\}$; // Restore all images with the ℓ-th AU
    end
    $\mathcal{D}_\ell \leftarrow \{\{(\mathbf{x}_i, Y_i)\}_{i=1}^{S} \in \mathcal{D} | Y_i^\ell = 1\}$ // Sample S (image,label) pairs of the ℓ-th AU
    $\mathcal{D} \leftarrow \mathcal{D} \setminus \mathcal{D}_\ell$; // Update current dataset
    $\mathcal{B}_i \leftarrow \mathcal{B}_i \cup \mathcal{D}_\ell$; // Update current batch
    Compute n_ℓ ($\ell = 1, ..., L$) as the number of the ℓ-th AU in $\mathcal{B}_i$;
    $\ell \leftarrow \arg\min_j n_j$; // Find the AU with fewest samples in $\mathcal{B}_i$;
  end
  Update N_ℓ ($\ell = 1, ..., L$) as the number of the ℓ-th AU in $\mathcal{D}$;
end
return $\mathcal{B}$;

5.2. Multi-label minority oversampling majority undersampling (MOMU)

To the best of our knowledge, despite numerous studies on multi-label classification and deep learning, there is limited discussion on how class imbalance of multi-label data can be systematically addressed between and within batches. As we have observed in the previous section, both random sampling and multi-label stratification suffer from dramatic unbalanced distributions *within* each mini-batch. This drives us to the next strategy termed multi-label minority oversampling majority undersampling (MOMU).

Algorithm 2 summarizes the proposed multi-label MOMU strategy. For each batch, MOMU proceeds by progressively filling the (image,label) pairs in a greedy manner. Similar to multi-label stratification, as discussed in the previous section, MOMU starts by picking S images that contain the AU with the fewest samples in the population distribution (the AU distribution of an entire dataset). Because each image contains multiple labels, adding S images into the current batch can simultaneously increase the base rate for other AUs. These S samples are then removed from the dataset to ensure a maximal use of annotated data. In the next iteration, MOMU picks the AU with the fewest samples in the current batch, and then samples next S images (without replacement) that contain this particular AU. In this way, we ensure that the AU with the fewest samples can be always compensated through sampling. We repeat the procedure for the desired number of B batches until all batches are filled. Note that during sampling, it is likely that a particular minority class runs out of samples ($N_\ell < S$). In this case, we simply restore to the dataset with all images that contain AU ℓ, and then continue sampling images that contain this particular AU class. Because the images are added into each batch consecutively with guarantees to contain at least an active AU, the class distribution *between* batches will remain around a similar scale. More importantly, as we intentionally fill in images for the minority class, the class distribution *within* batches can be also controlled within a balanced range.

The bottom row of Fig. 6 illustrates the AU distribution after the multi-label MOMU. As can be seen, the number of AU presence between batches remains in similar scale, while the AU distribution *within* batches becomes much more balanced. As we will show in the subsequent evaluation, such balanced distribution consistently improves training performance as well as test performance in both within-dataset and between-dataset scenarios. To our knowledge, this could serve as one of the first attempts that address multi-label sampling for unbalanced datasets in the context of stochastic training. Although we will illustrate only performance on deep learning models, we believe the same idea can be applied to more models such as multi-label stochastic SVMs [50].

Comparison with existing methods: Recall that most literature consider strategies that involve either resampling or classifier adaptation/cost-sensitive learning. One interpretation of MOMU is its behavior as a hybrid of both. As in standard deep learning, augmentation for training data is often done through random cropping of the input image. From this perspective, MOMU takes the full advantage of both types of strategies by achieving resampling through sampling the minor classes in the image space, and cost-sensitive learning through balancing the contributions of different classes in the feature space.

5.3. Evaluation of different multi-label sampling strategies

In this section, we evaluate the effects of multi-label sampling strategies in terms of improvements in training and test performance. Following Section 4.2, we used a 10-fold data split protocol. Note that, to reflect the severity of class imbalance in more realistic datasets, we used larger, renewed GFT and BP4D+ datasets [44] in this section. In comparison with Section 4.4, GFT and BP4D+ brought 147 subjects (2.9× more) and 98 subjects (2.4× more), respectively.

5.3.1. Evaluation of training performance

Fig. 7 reports the training performance on the GFT dataset in terms of F1-score (y-axis) and the number of iterations (x-axis). Three sampling strategies, *i.e.*, standard random sampling, multi-label stratification, and multi-label MOMU, were evaluated. The reason we picked F1-score as the evaluation metric is because of its sensitivity in true positives, which we believe can closely describe human perception compared to accuracy-based measures. In other words, given a distribution skewed toward negative samples in each AU class, we believe humans are more sensitive about a model classifying correctly on a positive sample than a negative one. If an accuracy-based metric (*e.g.*, S-score or kappa [49], AUC, or accuracy) is used over skewed classes, one may not be able to distinguish the classifier's performance on top of the true positives (see also [51]). Having such metric is able to provide a more accurate description about performance of human's interest.

As can be seen the red curve in Fig. 7, standard random sampling (as used in most deep learning literature) suffers from unbalanced AU distribution. For notational convenience, we denote base rate for the ℓ-th AU as BR_ℓ. The performance of minority AUs, such as AUs 4 ($BR_4 = 3.4\%$) and 15 ($BR_{15} = 8.8\%$), remains rather low even during the training phase with 8000 iterations. Multi-label stratification, as indicated by the green curve, exhibits a relatively smoother training curve because each mini-batch contains similar amount of AU presence, which would help avoid the network favoring prediction on the negative samples. However, as can be seen, multi-label stratification only ends up with similar performance because the AU distribution *within* each mini-batch remains dramatically biases as discussed in the previous section. The MOMU strategy, as indicated by the blue curve, shows significant improvement for minority classes, including AUs 1 ($BR_1 = 8.8\%$), 2 ($BR_2 = 10.9\%$), 4 ($BR_4 = 3.4\%$), and 15 ($BR_{15} = 8.8$). Not surprisingly, the performance of major AUs did not decrease notably even though the samples in the majority classes were under-sampled. This is mainly due to the high redundancy of the video frames shown in spontaneous datasets. In all, as indicated by the F1 scores, the multi-label MOMU strategy effectively guides the network with reliable training for the multi-label AU data.

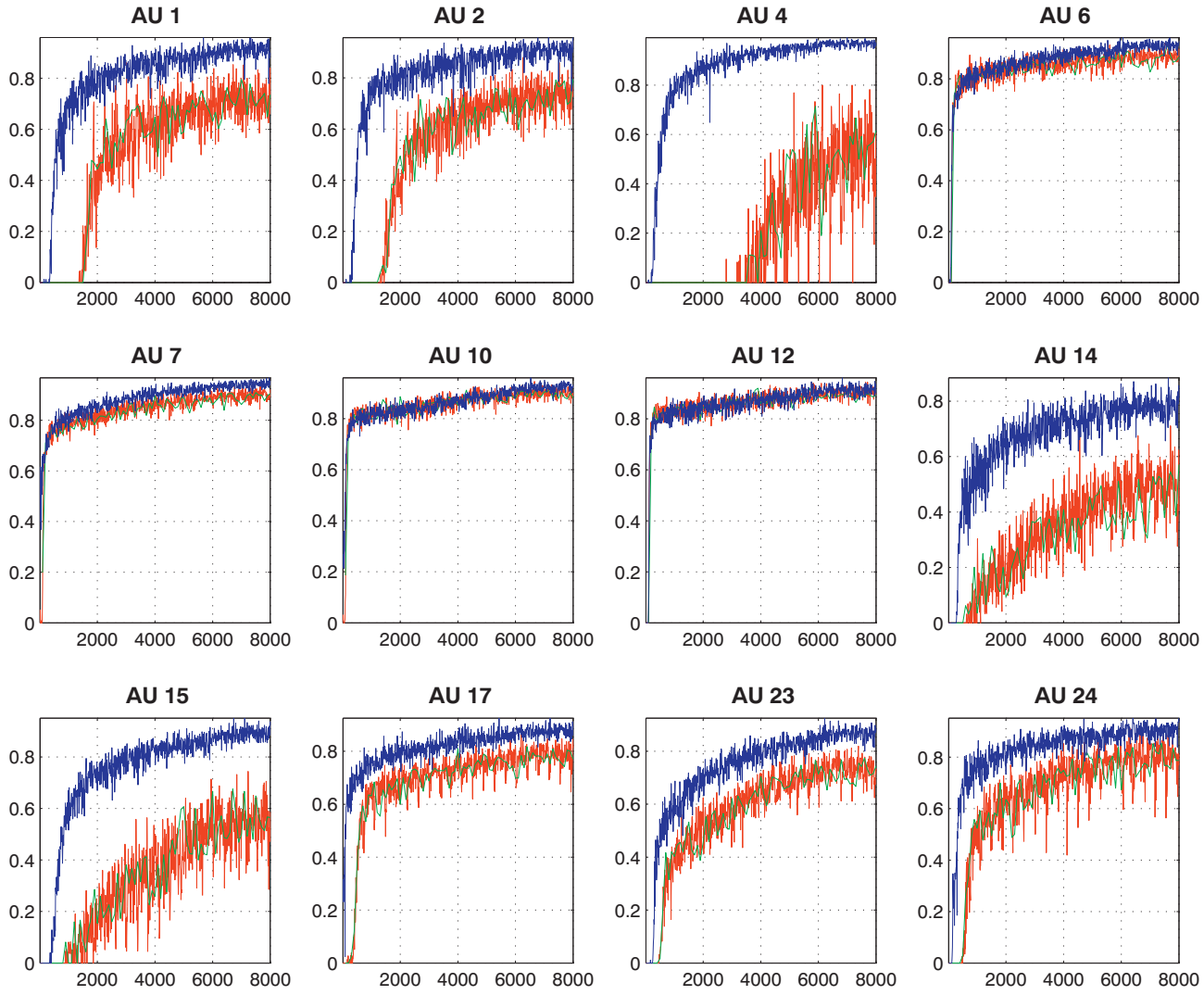


Fig. 7. Comparison of training performance on the GFT dataset in terms of F1-score (y-axis) vs the number of iterations (x-axis) over different sampling strategies: (red) random sampling, (green) multi-label stratification, (blue) multi-label MOMU. As can be observed, for conventional random sampling and multi-label stratification, the performance of minority AUs, such as AUs 4 ($BR_4 = 3.4\%$) and 15 ($BR_{15} = 8.8\%$), remains rather low even after training phase with 8000 iterations (the curve is higher better).

5.3.2. Evaluation of test performance

For performance evaluation during the test phase, we provide evaluation for individual AUs in Tables 3 and 4 in within- and between-dataset scenarios. The *within-dataset* scenario indicates training and test on subjects of the same dataset; *between-dataset* scenario indicates training on subjects of one dataset while test

Table 3
Performance evaluation of different sampling strategies: Random Sampling (RS), Multi-label stratification (MS), and MOMU sampling (MOMU). Comparison was performed in terms of within-dataset and between-dataset scenarios in the GFT dataset [35].

	Within			Between		
AU	RS	MS	MOMU	RS	MS	MOMU
1	.44	.38	.47	.03	.	.16
2	.41	.35	.38	.	.	.13
4	.	.	.29	.07	.07	.08
6	.73	.73	.71	.52	.52	.54
7	.72	.72	.73	.63	.61	.62
10	.68	.68	.67	.32	.33	.46
12	.72	.75	.75	.58	.61	.55
14	.05	.27	.4	.23	.25	.22
15	.17	.14	.29	.01	.01	.23
17	.32	.47	.49	.22	.3	.4
23	.39	.38	.48	.34	.43	.32
24	.13	.44	.41	.	.01	.19
Avg	.4	.44	.51	.25	.26	.32

Table 4
Performance evaluation of different sampling strategies: Random Sampling (RS), Multi-label stratification (MS), and MOMU sampling (MOMU). Comparison was performed in terms of within-dataset and between-dataset scenarios in the BP4D+ dataset [44].

	Within			Between		
AU	RS	MS	MOMU	RS	MS	MOMU
1	.19	.19	.43	.28	.31	.3
2	.15	.15	.46	.3	.32	.31
4	.83	.83	.88	.	.	.25
6	.82	.82	.9	.66	.78	.71
7	.91	.91	.94	.74	.86	.79
10	.78	.78	.82	.68	.74	.74
12	.87	.87	.91	.81	.75	.79
14	.8	.8	.83	.	.	.12
15	.15	.15	.38	.02	.05	.15
17	.3	.3	.54	.15	.01	.28
23	.44	.44	.6	.14	.17	.39
24	.02	.02	.4	.02	.	.09
Avg	.52	.52	.67	.25	.33	.41

on subjects of the other dataset. Fig. 8 shows the improvement on both GFT [35] and BP4D+ [44] datasets using within-dataset and between-dataset scenarios.

As can be seen in the *within-dataset* scenario of Fig. 8, the improvements on GFT focus on the minor classes, such as AUs 1, 2, 15, 17 and 24. More precisely, the improvements are mostly obvious in the F1-score metric. As mentioned earlier, this is because F1-score maintains the sensitivity in true positives, and therefore including more samples from the minority classes can help improve detection of the true positives. AUC did not reflect much improvement or decrement because of its insensitivity to skewed class distributions, as also discussed in [51]. On the other hand, the improvements within BP4D+ are rather consistent. One possible explanation is because BP4D+ yields more dramatic skewness between AU distributions than GFT does, and our multi-label MOMU strategy is able to better balance the distribution between and within batches. More interestingly, the improvements on BP4D+ are roughly inverse-proportional to the underlying AU base rates as shown in Fig. 5(b). This provides an evidence that training with a more balanced distribution in multi-label data can help improve test time performance, and the improvement is even more obvious when the class distributions are significantly different.

For the *between-dataset* scenario, the improvements of minority classes can be still observed for both datasets. Because BP4D+ has much higher base rate in AUs than BP4D does, AU 4 was significantly improved in the between-GFT experiment for both AUC and F1. For some AUs such as 1, 2, 6, 7, 10 and 12, the improvements were much less obvious. On the other hand, for the between-BP4D+ experiments, the results were rather mixed. For AUs 14, 17, 23 and 24, we observed similar behaviors. However, for AUs 1, 2, 6, and 7, AUC was improved yet F1 behaved in the opposite. Similarly, for AUs 10 and 12, the improvements in terms of AUC were higher than the ones in F1-score. One potential reason is because GFT has more subjects and thus more number of frames to train the classifier. Although within each AU the distribution is biased toward negative samples, having more training data can potentially improve prediction on negative samples, and thus improves AUC better. Nevertheless, multiple variabilities between two dataset can account for such relatively unpredictable results . These variabilities include recording environments, interview context, skin color, head pose and so on. We believe this is still an open problem, and refer interested readers to the Conclusion Chapter for more of our thoughts and ideas to address these variabilities.

6. Visualization of AU models

To better understand and interpret the proposed network, we implement a gradient ascent approach [53,54] to visualize each AU model. More formally, we look for such input image $\mathcal{I}^\star$ by solving the optimization problem:

$$\mathcal{I}^\star = \arg\max_{\mathcal{I}} A_\ell(\mathcal{I}) - \Omega(\mathcal{I}), \tag{3}$$

where $A_\ell(\mathcal{I})$ is an activation function for the ℓ-th unit of the output layer given an image $\mathcal{I}$, and $\Omega(\cdot)$ is a regularization function that penalizes $\mathcal{I}$ to enforce a natural image prior. In particular, we implemented $\Omega(\cdot)$ as a sequential operation of L_2 decay, clipping pixels with small norm, and Gaussian blur [54]. The optimization was done by iteratively updating a randomized and zero-centered image with the backprop gradient of $A_\ell(\mathcal{I})$. In other words, each pixel of $\mathcal{S}$ was renewed gradually to increase the activation of the ℓ-th AU. This process continued until 10,000 iterations.

Fig. 9 shows our visualizations of each AU model learned by the CNN architecture described in Section 3.1. As can be seen, most models match the attributes described in FACS [52]. For instance, model AU12 (lip corner puller) exhibits a strong "◡" shape to the mouth, overlapped with some vertical "stripes", implying the appearance of teeth is commonly seen in AU12. Model AU14 (dimpler) shows the dimple-like wrinkle beyond lip corners, which, compared to

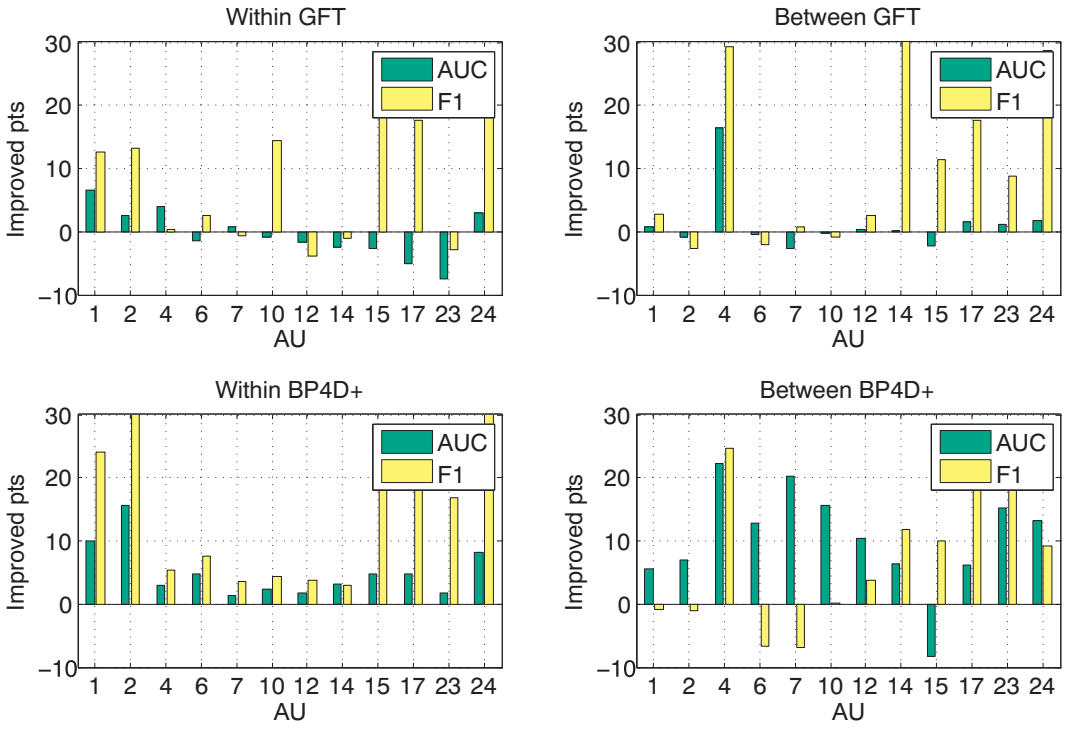


Fig. 8. Improved points of MOMU over random sampling in both within-dataset and between-dataset scenarios for GFT [35] and BP4D+ [44] datasets. Results in AUC and F1 suggest that improvements are more consistent in BP4D+ than in GFT due to the more dramatic AU imbalance in the BP4D+ dataset (as illustrated in Fig. 5).

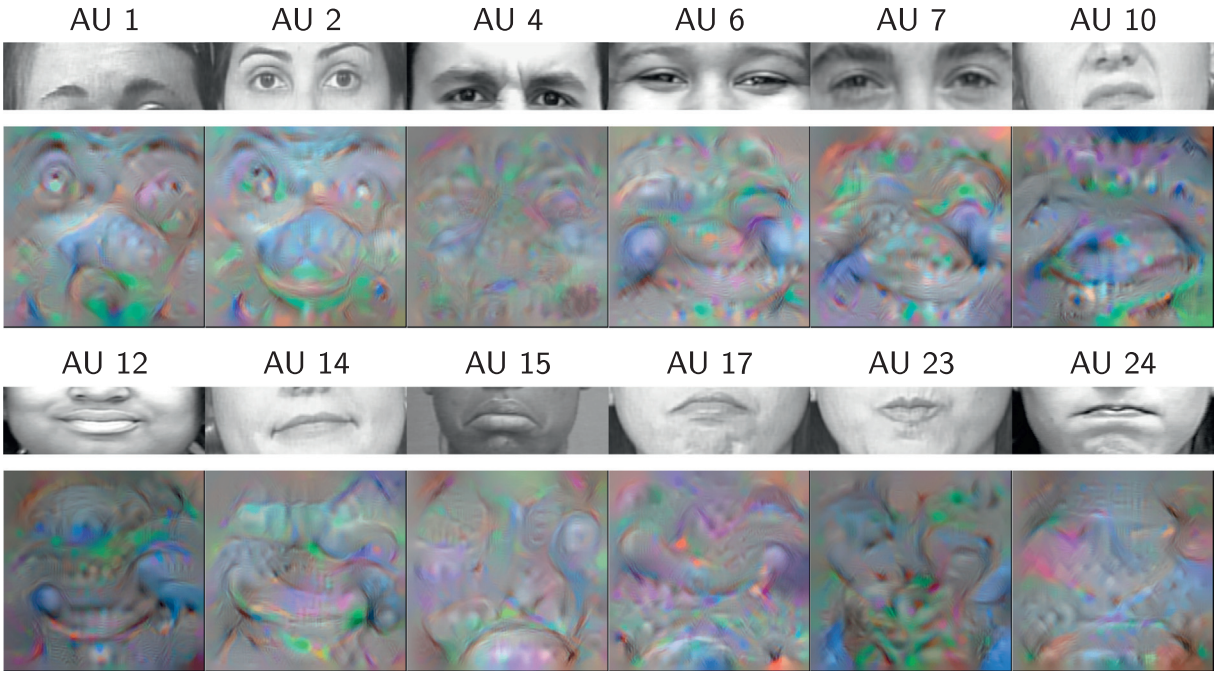


Fig. 9. Synthetically generated images to maximally activate individual AU neurons in the output layer of CNN, trained on GFT [35], showing what each AU model "wants to see". The learned models show high agreement on attributes described in FACS [52] (best view electronically).

AU12, gives the lip corners a downward cast. Model AU15 (lip corner depressor) shows a clear "⌣" shape to the mouth, producing an angled-down shape at the corner. For upper face AUs, model AU6 (cheek raiser) captures deep texture of raised-up cheeks, narrowed eyes, as well as a slight "⌣" shape to the mouth, suggesting its frequent co-occurrence with AU12 in spontaneous smiles. Models AU1 and AU2 (inner/outer brow raiser) both capture the arched shapes to the eyebrows, horizontal wrinkles above eyebrows, as well as the widen eye cover that are stretched upwards. Model AU4 (brow lowerer) captures the vertical wrinkles between the eyebrows and narrowed eye cover that folds downwards.

Our visualizations suggest that the CNN was able to identify these important spatial cues to discriminate AUs, even though we did not ask the network to specifically learn these AU attributes. Furthermore, the global structure of a face was actually preserved throughout the network, despite that convolutional layers were designed for local abstraction (*e.g.*, corners and edges as shown in Fig. 2(d)). Lastly, the widespread agreements between the synthetic images and FACS [52] confirm that the learned representation is able to describe and reveal co-occurring attributes across multiple AUs. We believe such AU co-occurrence is captured due to the multi-label structure in the proposed network. This was not shown possible in standard hand-crafted features in AU detection (*e.g.*, shape [28,55], HOG [12,14], LBP [21,56] or Gabor [21]). To the best of our knowledge, this is the first time to visualize how machines see facial AUs.

7. Conclusion and future work

We have presented a hybrid network that jointly learns three factors for multi-label AU detection: *Spatial representation*, *temporal modeling*, and *AU correlation*. To the best of our knowledge, this is the first study that shows a possibility of learning the three seemingly unrelated aspects within one framework. The hybrid network is motivated by existing progress on deep models, and takes advantage of spatial CNNs, temporal LSTMs, and their fusions to achieve multi-label AU detection. Experiments on two large spontaneous AU datasets demonstrate the performance over a standard CNN and feature-based state-of-the-art methods. In addition, we introduce multi-label sampling strategies to further improve performance for sparse AUs. Lastly, our visualization of learned AU models showed, for the first time, how machines see each facial AU. Models trained with the sampling strategies showed promising improvements on both validation and test data. Future work include deeper analysis of the temporal network (*e.g.*, evaluate the impact of head movement of pose for temporal modeling, and incorporation of bi-directional LSTMs), training an entire network end-to-end, and compare the proposed model between single-label and multi-label settings, etc.

Acknowledgment

This work was supported in part by the US National Institutes of Health grants GM105004 and MH096951 and Division of Computer and Network Systems grant number 1629716. The authors also thank NVIDIA for supporting this research with a Tesla K40c GPU, and Jiabei Zeng and Kaili Zhao for assisting partial experiments.

References

[1] P. Ekman, W. Friesen, J.C. Hager, Facial action coding system, A Human Face, 2002.
[2] P. Ekman, E. Rosenberg, What the Face Reveals, 2nd ed., 2005.
[3] J.F. Cohn, F. De la Torre, Automated face analysis for affective, The Oxford Handbook of Affective Computing, 2014, 131.
[4] W.-S. Chu, F. De la Torre, J.F. Cohn, Selective transfer machine for personalized facial action unit detection, CVPR, 2013.
[5] E. Sangineto, G. Zen, E. Ricci, N. Sebe, We are not all equal: personalizing models for facial expression analysis with transductive parameter transfer, ACM MM, 2014.
[6] S. Yang, O. Rudovic, V. Pavlovic, M. Pantic, Personalized modeling of facial action unit intensity, Advances in Visual Computing, Springer. 2014, pp. 269–281.
[7] W.-S. Chu, F. De la Torre, J.F. Cohn, Learning spatial and temporal cues for multi-label facial action unit detection, AFGR, 2017.
[8] F. De la Torre, J.F. Cohn, Facial expression analysis, Visual Analysis of Humans: Looking at People, 2011, 377.

[9] A. Martinez, S. Du, A model of the perception of facial expressions of emotion by humans: research overview and perspectives, J. Mach. Learn. Res. 13 (2012) 1589–1608.
[10] M.F. Valstar, M. Mehu, B. Jiang, M. Pantic, K. Scherer, Meta-analysis of the first facial expression recognition challenge, IEEE Trans. Syst. Man Cybern. B Cybern. 42 (4) (2012) 966–979.
[11] E. Sariyanidi, H. Gunes, A. Cavallaro, Automatic analysis of facial affect: a survey of registration, representation, and recognition, IEEE Trans. Pattern Anal. Mach. Intell. 37 (6) (2015) 1113–1133.
[12] J. Zeng, W.-S. Chu, F. De la Torre, J.F. Cohn, Z. Xiong, Confidence preserving machine for facial action unit detection, ICCV, 2015.
[13] O. Rudovic, V. Pavlovic, M. Pantic, Context-sensitive dynamic ordinal regression for intensity estimation of facial action units, IEEE Trans. Pattern Anal. Mach. Intell. 37 (5) (2015) 944–958.
[14] K. Zhao, W.-S. Chu, F. De la Torre, J.F. Cohn, H. Zhang, Joint patch and multi-label learning for facial action unit detection, CVPR, 2015.
[15] L. Zhong, Q. Liu, P. Yang, B. Liu, J. Huang, D.N. Metaxas, Learning active facial patches for expression analysis, CVPR, 2012.
[16] K.-Y. Chang, T.-L. Liu, S.-H. Lai, Learning partially-observed hidden conditional random fields for facial expression recognition, CVPR, 2009.
[17] J. Chen, M. Kim, Y. Wang, Q. Ji, Switching Gaussian process dynamic models for simultaneous composite motion tracking and recognition, CVPR, 2009.
[18] P. Liu, S. Han, Z. Meng, Y. Tong, Facial expression recognition via a boosted deep belief network, CVPR, 2014.
[19] S. Du, A.M. Martinez, Compound facial expressions of emotion: from basic research to clinical applications, Dialogues Clin. Neurosci. 17 (4) (2015) 443.
[20] Y. Tong, W. Liao, Q. Ji, Facial action unit recognition by exploiting their dynamic and semantic relationships, IEEE Trans. Pattern Anal. Mach. Intell. 29 (10) (2007) 1683–1699.
[21] Z. Wang, Y. Li, S. Wang, Q. Ji, Capturing global semantic relationships for facial action unit recognition, ICCV, 2013.
[22] S. Eleftheriadis, O. Rudovic, M. Pantic, Multi-conditional latent variable model for joint facial action unit detection, ICCV, 2015.
[23] C. Wu, S. Wang, Q. Ji, Multi-instance Hidden Markov Model for facial expression recognition, AFGR, 2015.
[24] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, NIPS, 2012.
[25] K. Zhao, W.-S. Chu, H. Zhang, Deep region and multi-label learning for facial action unit detection, CVPR, 2016.
[26] A. Gudi, H.E. Tasli, T.M. den Uyl, A. Maroulis, Deep learning based FACS action unit occurrence and intensity estimation, AFGR, 2015.
[27] S. Ghosh, E. Laksana, S. Scherer, L.-P. Morency, A multi-label convolutional neural network approach to cross-domain action unit detection, ACII, 2015.
[28] S. Jaiswal, M.F. Valstar, Deep learning the dynamic appearance and shape of facial action units, WACV, 2016.
[29] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, NIPS, 2014.
[30] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, X. Xue, Modeling spatial-temporal clues in a hybrid deep learning framework for video classification, ACM MM, 2015.
[31] J. Donahue, L.A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, T. Darrell, Long-term recurrent convolutional networks for visual recognition and description, CVPR, 2015.
[32] X. Zhang, L. Yin, J.F. Cohn, S. Canavan, M. Reale, A. Horowitz, P. Liu, A high-resolution spontaneous 3D dynamic facial expression database, AFGR, 2013.
[33] A. Graves, A.-r. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, ICASSP, 2013.
[34] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, CVPR, 2014.
[35] J.F. Cohn, M.A. Sayette, Spontaneous facial expression in a small group can be automatically measured: an initial demonstration, Behav. Res. Methods 42 (4) (2010) 1079–1086.
[36] F. De la Torre, W.-S. Chu, X. Xiong, F. Vicente, X. Ding, J.F. Cohn, IntraFace, Automatic Face and Gesture Recognition, 2015.
[37] X. Ding, W.-S. Chu, F. De la Torre, J.F. Cohn, Q. Wang, Facial action unit event detection by cascade of tasks, IEEE Conference on International Conference on Computer Vision, 2013.
[38] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional Architecture for Fast Feature Embedding, arXiv preprint. (2014) arXiv:1408.5093.
[39] O.M. Parkhi, A. Vedaldi, A. Zisserman, Deep face recognition, British Machine Vision Conference, 2015.
[40] J. Lin, Divergence measures based on the Shannon entropy, IEEE Trans. Inf. Theory 37 (1) (1991) 145–151.
[41] A. Wong, M. You, Entropy and distance of random graphs with application to structural pattern recognition, IEEE Trans. Pattern Anal. Mach. Intell. (5) (1985) 599–609.
[42] Y. Zhu, F. De la Torre, J.F. Cohn, Y.-J. Zhang, Dynamic cascades with bidirectional bootstrapping for spontaneous facial action unit detection, IEEE Trans. Affect. Comput. 2 (2011) 79–91.
[43] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Web-scale training for face identification, CVPR, 2015.
[44] X. Zhang, L. Yin, J.F. Cohn, S. Canavan, M. Reale, A. Horowitz, P. Liu, J.M. Girard, Bp4d-spontaneous: a high-resolution spontaneous 3d dynamic facial expression database, Image Vis. Comput. 32 (10) (2014) 692–706.
[45] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.
[46] R.C. Prati, G. Batista, D.F. Silva, Class imbalance revisited: a new experimental setup to assess the performance of treatment methods, Knowl. Inf. Syst. 45 (1) (2015) 247–270.
[47] Y. Sun, A. Wong, M.S. Kamel, Classification of imbalanced data: a review, Int. J. Pattern Recognit. Artif. Intell. 23 (04) (2009) 687–719.
[48] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.
[49] J. Girard, W.-S. Chu, L. Jeni, J.F. Cohn, F. De la Torre, Sayette Group Formation Task (GFT): spontaneous facial expression database, AFGR, 2017.
[50] M. Lapin, M. Hein, B. Schiele, Loss functions for top-k error: analysis and insights, CVPR, 2016.
[51] T. Fawcett, An introduction to ROC analysis, Pattern Recogn. Lett. 27 (8) (2006) 861–874.
[52] P. Ekman, E.L. Rosenberg, What the Face Reveals: Basic and Applied Studies of Spontaneous Expression Using the Facial Action Coding System (FACS), Oxford University Press, USA, 1997.
[53] K. Simonyan, A. Vedaldi, A. Zisserman, Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, arXiv preprint. (2013) arXiv:1312.6034.
[54] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, H. Lipson, Understanding Neural Networks Through Deep Visualization, 2015.
[55] P. Lucey, J.F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, I. Matthews, The Extended Cohn-Kanade Dataset (CK+): a complete dataset for action unit and emotion-specified expression, CVPR Workshops, 2010.
[56] B. Jiang, M.F. Valstar, M. Pantic, Action unit detection using sparse appearance descriptors in space-time video volumes, AFGR, 2011.