2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

Hybrid Hypergraph Construction for Facial Expression Recognition

Yuchi Huang* and Hanqing Lu†
National Lab of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
95 Zhongguancun East Road, 100190, Beijing, China
Email: *yuchi.huang@foxmail.com, †luhq@nlpr.ia.ac.cn,

***Abstract*—In this paper, we proposed a novel framework for facial expression recognition, in which face images were taken as vertices in a hypergraph and the task of expression recognition was formulated as the problem of hypergraph based inference. A hybrid strategy was developed to construct hyperedges: we generated probabilities of facial action units by deep convolutional networks and took each action unit as an 'attribute' to represent a hyperedge; we also formed hyperedges by using embedded network features before the last full connected layer to perform local clustering. In this way, each face image was assigned to various hyperedges by exploiting the representational power of deep convolutional networks. Our facial expression recognition system generates expression labels by a hypergraph based transductive inference approach, which tends to assign the same label to vertices that share many incidental hyperedges, with the constraints that predicted labels of training images should be similar to their ground truth labels. We compared the proposed approach to state-of-the-art methods and its effectiveness was demonstrated by extensive experimentation.**

I. Introduction

Facial expressions are vital to social communication between humans. Although humans can visually recognize facial expressions easily, it is quite a big challenge for machines to interpret this information. Earlier work [1] [2] in this area is mostly based on the Facial Action Coding System (FACS) [3] proposed by Paul Ekman, in which facial expressions were decomposed into 'action units' (AUs), i.e. movements of face regions such as human's eyes, nose and month. These AU-based methods are very dependent on carefully hand-engineered features to ensure good performance. Another category of work for expression analysis used human's general appearance information such as facial shape and texture features to model a person's expression [4] [5] [6]. Due to the success of deep learning in various computer vision problems, recent work in this category applied deep convolutional neural networks (D-CNN) as appearance-based classifier to detect action unit occurrence [7] or expression classes [8] [9] and achieved encouraging results. The common ground of [7] [8] and [9] is that they all trained an end-to-end D-CNN system to predict AU/expression classes.

In this paper, we combined the best of two worlds by utilizing both the action unit information and appearance features in a hypergraph framework to solve the problem of facial expression recognition. As defined in [10], a *hypergraph* is a graph in which an edge (i.e. hyperedge) can connect more than two vertices. That is, a hypergraph is a generalization of a pairwise simple graph, in which a set of vertices that have the same 'attribute' or belong to the same local cluster is defined as a weighted hyperedge; the magnitude of a hyperedge weight indicates to what extent the vertices in a hyperedge belong to the same cluster [11]. A vetex in a hypergraph may belong to various hyperedges. The hypergraph model has proven to be beneficial to various clustering/classification tasks [12] [13] [14] [15], because it can represent the information that three or more vertices have the same semantic attribute, which usual graphs can not describe. However, previous hypergraph frameworks usually employed SIFT-like descriptors [16] for feature extraction and hyperedge construction [17], which may limit the potential of hypergraphs in real-world applications. In this paper, we adopted the deep convolutional networks to predict the occurrence of action units and to generate the appearance features. Both two kinds of information were used for hyperedge construction respectively.

As shown in Figure 1, our expression recognition pipeline consists of two stages. Inspired by the observation of [8] that the hidden units of high-level layers in deep convolutional neural networks (D-CNN) resemble facial action units or combinations of them, in the first stage a D-CNN model was built and its softmax layer was formulated to generate the probabilities of different action units. Naturally, each action unit represents a facial attribute critical to expression classification.

To utilize the information generated by our D-CNN model, in the second stage a hybrid strategy was developed to construct two kinds of hyperedges. On the one hand, each action unit was taken as an 'attribute' to represent a hyperedge. Considering that each action unit output computed in the first stage must represent multiple facial images, it is natural to describe the relationship among samples in a facial expression dataset as a hypergraph; a face image can be assigned to corresponding hyperedges according to its output probabilities on various action units. On the other hand, to further exploit the correlation information among expression images, we also formed hyperedges by using embedded network features before the last full connected layer to perform local clustering.

978-1-5090-4847-2/16/$31.00 ©2016 IEEE

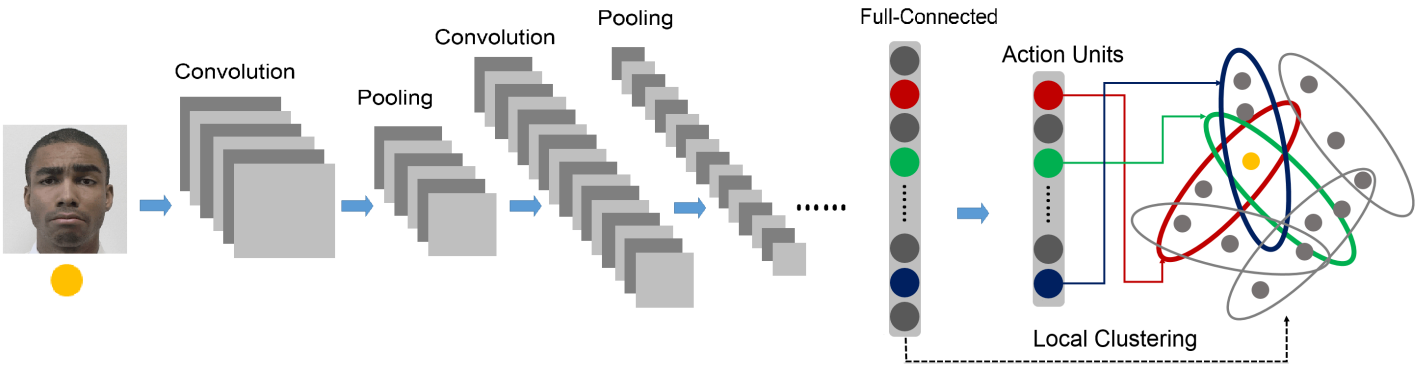

Fig. 1. The framework of hybrid hypergraph construction for facial expression Recognition, in which all face images in a dataset can be processed by our trained deep convolutional neural networks (D-CNN) to build a hypergraph. The last softmax layer of the D-CNN model was formulated to generate the probabilities of different action units. Each action unit was used as a critical facial attribute to create a hyperedge. To further exploit the correlation information among expression images, we also formed hyperedges by using embedded network features to perform local clustering: we took each facial image as a 'centroid' vertex and formed a hyperedge by a centroid and its k-nearest neighbors. Take the input image (illustrated as a yellow vertex) in this figure as an example, three action unit attributes (illustrated as red, green and blue nodes) are activated and the yellow vertex is assigned to three corresponding hyperedges. It is also taken as a 'centroid' vertex to form a hyperedge by utilizing its embedded network feature.

Based on the similarities computed from embedded D-CNN features, we took each facial image as a 'centroid' vertex and formed a hyperedge by a centroid and its k-nearest neighbors. We followed the fuzzy hypergraph model introduced in [17], which presents not only whether a vertex v_i belongs to a hyperedge e_j, but also the probability that $v_i \in e_j$. Based on the built hypergraph, our facial expression recognition system generates class labels by a transductive inference approach, which tends to assign the same label to vertices (images) that share many incidental hyperedges (attributes or local clusters), with the constraints that predicted labels of training images should be similar to their ground truth labels. We compared the proposed approach to state-of-the-art methods and its effectiveness was demonstrated by extensive experimentation.

The rest paper is organized as follows: The definition to hypergraph is introduced in Section 2; we address the deep learning based hyperedge constructionin in Section 3, and we present the hypergraph learning for expression recognition in Section 4; Experiments are reported in Section 5, and followed by the conclusion finally.

II. Revist to Hypergraph

Let V represent a finite set of vertices and E a family of subsets of V such that $\bigcup_{e\in E} = V$. $G = (V, E, w)$ is called a hypergraph with the vertex set V and the hyperedge set E, and each hyperedge e is assigned a positive weight $w(e)$. A traditional hypergraph can be represented by a $|V| \times |E|$ incidence matrix H_t:

$$h(v_i, e_j) = \begin{cases} 1, & \text{if } v_i \in e_j \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Such a representation with higher order relationships has illustrated the benefits to various clustering/classification tasks [14] [17], because it takes account of the relationship not only between two vertices, but also among three or more vertices containing local grouping information, which simple pairwise graphs can not describe. However, as illustated in Equation 1, this structure assigns a vertex v_i to a hyperedge e_j with a binary decision, which causes some information loss. In this paper, we adopted a fuzzy version of hypergraph proposed in [17] to overcome this limitation by assign each vertex v_i to a hyperedge e_j according to the probability of v_i belonging to e_j:

$$h(v_i, e_j) = \begin{cases} O(j,i), & \text{if } O(j,i) > t \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where $O(j, i)$ denotes the probability of a face image i having an attribute or belonging a local cluster j, which will be explained in the next section; t is a threshold. Here each attribute/local cluster is corresponding to a hyeredge. According to this formulation, v_i is 'softly' assigned to e_j based on the computed probability $O(j, i)$. For a vertex $v \in V$, its degree is defined to be $d(v) = \sum_{e\in E} w(e)h(v, e)$. For a hyperedge $e \in E$, its degree is defined as $\delta(e) = \sum_{v\in e} h(v, e)$. D_v, D_e and W are used to denote the diagonal matrices of the vertex degrees, the hyperedge degrees and the hyperedge weights respectively.

$\#of AUs$

III. Hybrid Hyperedge Construction

A. Our Deep Convolutional Neural Networks

Network architecture. In this paper we constructed the deep convolutional networks as in Table I. The architecture

	1	2	3	4	5	6	7	8	9	10
Layer	Conv.	Pool.	Conv.	Conv.	Pool.	Conv.	Pool.	Full Conn.	L2	Loss
Kernel size on each channel	3×3	3×3	3×3	3×3	3×3	3×3	3×3	–	–	–
# of feat. maps or feat. dims	96	–	128	128	–	192	–	512	512	–

TABLE I
STRUCTURE OF OUR DEEP CONVOLUTIONAL NEURAL NETWORKS. THE LAST LINE SHOWS THE NUMBER OF FEATURE MAPS FOR CONVOLUTIONAL LAYERS AND FEATURE DIMENSIONS FOR FULL CONNECTED LAYERS.

consists of four convolutional layers containing 96, 128, 128 and 192 filters, respectively. Except those hidden units in the first layer (inputs of which are 49×49 gray-level images), all other convolutional layers use 3D convolutional filters. We adopted the rectified linear unit (ReLU) as the activation function. We used two consecutive convolutional layers (Layer 3 and Layer 4) to increase the representational power. The size of all filters on each feature map is 3×3 to capture more detailed texture variation. We used 3×3 pooling with stride 2 for all pooling layers. The last convolutional layer is followed by a full-connected layer containing 512 hidden units. We implemented this D-CNN model on the famous deep learning library CAFFE created by Jia [18].

Training Protocols. We trained our models using stochastic gradient descent with a batch size of 128 examples; we set momentum = 0.9 and weight decay = 0.001; we initialized all the weights from zero-mean Gaussian distribution with a standard deviation 0.005. We also used dropout [19] and various forms of data augmentation to regularize our networks and reduce overfitting. We applied dropout to all the convolutional layers and fully-connected layers with a probability of 0.5. For data augmentation, we applied the following transformations to each image: translations, horizontal flips, rotations, scaling and pixel intensity augmentation.

Pre-training and fine-tuning. We discriminatively pre-trained our D-CNN on the Labeled Faces in the Wild dataset (LFW) [20] which contains 5,749 subjects and 13,233 cropped face images. During this phase we used 4,000 output corresponding to the number of selected training subjects till the networks converge. Then we replaced the 4000-way classification with randomly initialized N-way classification, where N is the number of facial action units presented in a specific expression dataset. Stochastic gradient descent training of the D-CNN parameters was continued by using the corresponding expression dataset. We found the pre-training and followed domain-specific fine-tuning is very effective to boost performance of our task. So far we have built a D-CNN model which is able to predict the occurrence of an action unit directly.

B. Hyperedge Construction using Action Units

Each action unit contains specific semantic information which is beneficial to the expression recognition task. We took action units as facial expression 'attributes' and used them to construct hyperedges. During this stage, the output layer of our trained D-CNN is designed to predict the occurrence of N action units. The output of last full connected layer is exponientially normalized in the softmax layer as follows:

$$O(j,i) = \frac{e^{C(i,j)}}{\sum_{k=1}^{N} e^{C(i,k)}}, \tag{3}$$

where $C(i,j)$ denotes the value of jth input unit of the softmax layer for image i, N represents the number of action units. In this way each output unit here represents a **hyperedge** and the term of $O(j,i)$ is defined as the probability of image i contains action unit j. During the training phase, the training face images and corresponding action units labels are used to fine-tune our D-CNN model. During the testing phase, both the training and testing samples of a dataset are input into the D-CNN model and N hyperedges can be constructed according to $O(j,i)$. Note that t in Equation 2 is empirically set to the average of all positive output node values of the softmax layer.

The hyperedge weight $w(e_i)$ is computed as follows:

$$w(e_j) = \sum_{v_i \in e_j} O(j,i). \tag{4}$$

Based on this definition, a hyperedge (action unit) is more important if the total probabilities (of the images it contains) is higher.

C. Hyperedge Construction via Local Clustering

Deep convolutional neural networks are often employed as a feature extraction/embedding tool and achieve state-of-the-art performance in various applications [21]. In this paper, we adopted the feed-forward network features $G(I)$ of the 8th layer in Table I to compute the affinities among facial expression images, which we believe can provide effective complementary information for the expression recognition task. To reduce the sensitivity to illumination changes, we followed the method of [22] to normalize this 512-dimentional feature $G(I)$:

$$\bar{G}(I)_i = \frac{G(I)_i}{max(G_i, \epsilon)}, \tag{5}$$

$$f(I) := \frac{\bar{G}(I)}{\|\bar{G}(I)\|_2}, \tag{6}$$

where $\epsilon = 0.05$ in order to avoid division by a small number. In Equation 5 each component of the feature vector is divided by its largest value across the training set; in Equation 6 an L2-normalization is performed. Based on normalized features, we can form a $|V| \times |V|$ affinity matrix A over the image

set V computed based on some measurement and $A(i,j) \in [0,1]$. We simply chose the inner product between the two normalized feature vectors as the similarity measurement.

As in [17], we exploited the correlation among expression images by local clustering. We took each vertex as a exemplar 'centroid' vertex and formed a hyperedge by a centroid and its k-nearest neighbors. That is, the size of such a hyperedge is $k+1$. The probability of a face image i belonging a local cluster j is defined as follows:

$$O(j,i) = \begin{cases} A(i,j), & \text{if } v_i \in e_j \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

According to this formulation, a vertex v_i is 'softly' assigned to e_j based on the similarity $A(i,j)$ between v_i and v_j, where v_j is the centroid of e_j. This hyperedge structure presents the local grouping information which may beneficial to our expression recognition task. Considering that an face image represents an 'exemplar' expression and in the feature space its k nearest neighbors are very similar to it, the probability that these $k+1$ images contains the same expression is high. By applying this hyperedge constructed from local clustering, the relationship among different facial expression images is more completely described.

Intuitively, a small size hyperedges only contain 'micro-local' grouping information which will not help the global clustering over all the images, and very large-size hyperedges may contain images from different classes and suppress diversity information. In this work we performed a sweep over all the possible k values of the hyperedge size to optimize the classification results.

The hyperedge weight $w(e_i)$ is also computed as Equation 4. This is natural: a 'compact' hyperedge (local cluster) with higher inner group similarities is assigned a higher weight, which means compact hyperedges are more important.

IV. Hypergraph Learning

In the classical work of hypergraph learning, the normalized cost function [10] $\Omega(f)$ of a bi-partition problem is defined as follows:

$$\begin{aligned} \Omega(f) &= \frac{1}{2}\sum_{e\in E}\sum_{u,v\in e}\frac{w(e)h(u,e)h(v,e)}{\delta(e)}\left(\frac{f(u)}{\sqrt{d(u)}} - \frac{f(v)}{\sqrt{d(v)}}\right)^2 \\ &= f^T(I-\Theta)f, \end{aligned} \tag{8}$$

where the vector f is the image labels to be learned; $\Theta = D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}}$ and $I-\Theta$ is a positive semi-definite matrix called the hypergraph Laplacian. By minimizing this cost function, images sharing many incidental hyperedges are guaranteed to obtain similar labels. In [17], it is verified that derivation in Equation 8 also holds for the fuzzy hypergraph. In an unsupervised framework, Equation 8 can be optimized by the eigenvector related to the smallest nonzero eigenvalue of $I-\Theta$.

In a two-class transductive learning setting [10], a vector y can be defined to represent initial labeling information: $y(v) = \frac{1}{|Pos|}$, if a vertex v is in the positive training set Pos, $y(v) = -\frac{1}{|Neg|}$, if it is in the negative training set Neg. If v is in the testing set or unlabeled, $y(v) = 0$. To force the learned labels to approach the initial labeling y, a regularization term is defined as follows:

$$\|f-y\|^2 = \sum_{u\in V}(f(u)-y(u))^2. \tag{9}$$

After this regularization term is introduced, the learning task is to minimize the sum of above two cost terms with respect to f, which is

$$\Phi(f) = f^T\Delta f + \mu\|f-y\|^2, \tag{10}$$

where $\mu > 0$ is the regularization parameter. Differentiating $\Phi(f)$ with respect to f, we have

$$f = (1-\gamma)(I-\gamma\Theta)^{-1}y, \tag{11}$$

where $\gamma = \frac{1}{1+\mu}$. This is equivalent to solving the linear system $((1+\mu)I - \Theta)\, f = \mu y$.

In our application, we constructed a hypergraph for all facial expression images with M different initial labeling vectors y, where M is the number of expressions present in a specific dataset. In each of these labeling vectors a positive/negative label denotes the presence/non-presence of one expression on a training sample; an initial label 0 denotes that the corresponding image is in the testing set. With M different ys above linear system will be solved for M times and the final learned expression of a test image is decided by the maximum value of M predicted scores.

V. Experiments

We chose two representative facial expression datasets in our experiments: the extended Cohn-Kanade database (CK+) [23] and the dataset for Facial Expression Recognition Challenge 2013 (FER2013) [24]. We compared our proposed method to two baseline approaches: 1) D-CNN + SVM in which normalized features $f(I)$ of the first full-connected layer (as in Equation 6) are fed to a linear SVM classifier and 2) D-CNN + M-way Softmax in which M expression labels are predicted directly (by using the configuration of M-way expression classification in the last two layers). We also compared to the state-of-the-art methods on each dataset to show the advantage of our approach.

For the hyperedges formed by action units, the hyperedge size is decided by Equation 2. For the hyperedges formed by local clustering, as introduced in Section III-C, we performed a sweep over all the possible k values and used the respective optimal hyperedge size 30 in all experiments. For the parameter γ in Equation 11, we followed the original work of

Zhou [25] and fix it as 0.1 for the best performance. Other parameters were directly computed from experimental data.

A. Performance on CK+

The extended Cohn-Kanade database (CK+) [23] contains 593 sequences across 123 subjects which are all FACS coded at the last frame. All sequences are from the neutral face to the peak expression. For each sequence the corresponding action units and their intensities of the last frame are labeled by certified FACS coders. ONLY 327 of the 593 sequences have emotion labels and each of which is assigned one of 7 expressions: Anger, Contempt, Disgust, Fear, Happy, Sad and Surprise.

To build the expression dataset for comparison of results, we followed the protocol of [26] [8] to extract last three frames of 327 sequences with expression labels. We also followed [26] [8] to use the first frame of each sequence as a neutral expression. In this way we formed a set of 1308 images with 8 different expression labels.

To train our deep neural networks, at first we pre-trained our model as described in Section III-A. Since detected faces in CK+ images are all frontal, we used frontalized LFW data provided by [27] for the pre-training. To fulfil the fine-tuning, we only utilized those 266 sequences in which action units are labeled but expressions are not. We extracted last five frames of these 266 sequences to form a set of 1330 images, applied face detection and rescaled them into 49×49 images. In our experiments, we adopted a subset of $N = 24$ labeled action units (as shown in Table II) because only the frequencies of those action units are adequate for model fine-tuning. In this paper, only the presence/absence (1/0) information of the action units were used as ground truth and the intensities were neglected. Data augmentation techniques were utilized to optimize the tuning results. So far we have built a D-CNN model which is able to predict the occurrence of an action unit directly; the average prediction accuracy of 24 action units on the expression set of 1308 images (defined above) is 96.9%.

Based on the action unit occurrence output of trained D-CNN model, we constructed 24 hyperedges by action units and 1308 hyperedges by local culstering. We then split 1308 images into 10 subject independent subsets in the manner presented by [26] and performed 10 fold cross-validation. As described in Section IV, in each experiment 8 different ys w.r.t. 8 expressions were used and the final learned expression was decided by the maximum value of 8 predicted scores.

By using the same D-CNN model, our hypergraph learning approach outperforms D-CNN + SVM and D-CNN + M-way Softmax by 2.1% and 2.9%, respectively. Results of all these three approaches are also significantly better than those of reported by start-of-the-art methods [28] [8].

B. Performance on FER2013

FER2013 consists of 28,709 48×48 training, 3,589 validation and 3,589 testing images of faces under 7 different types of expression: Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral. This dataset was used for ICML 2013 Challenge for Representation Learning. As shown in Figure 2, the face images of this dataset are not frontalized. So without frontalization we only rescaled the detected LFW faces to 49×49 and used them for pre-training of our D-CNN model mentioned in Section III-A. Then all above CK+ images ($1308 + 1330$ images) with action unit labels were used in the fine-tuning. To achieve better results on FER2013, we augmented the CK+ data by introducing large in-plane rotation up to $\pm 45°$. Then all the images in FER2013 were padded to 49×49 and were input to our D-CNN model to form a hypergraph as introduced in Section IV. The training and validation images were used as the 'training' set to get 7 different initial labeling vectors. As illustrated in Table IV, our approach overperforms DLSVM [9] (the winning method of ICML 2013 Challenge for Representation Learning) by 6.2%.

Fig. 2. Training data of FER2013. Each row consists of faces of the same expression: starting from the first row: Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral.

VI. Conclusion

We introduced a transductive learning framework for facial expression recognition, in which fuzzy hypergraph was utilized to represent the relevance relationship among the images. Using output probabilities of action units and embedded features produced by our D-CNN model, we took each image as a vertex and formed hyperedges according to those deep learning driven semantic attributes. In this way, the task of facial expression classification was converted to a transductive learning problem which can be solved by the hypergraph partition algorithm. The effectiveness of our proposed method was

AU	Name	AU	Name	AU	Name	AU	Name
1	Inner Brow Raiser	9	Nose Wrinkler	16	Lower Lip Depressor	25	Lips Part
2	Outer Brow Raiser	10	Upper Lip Raiser	17	Chin Raiser	26	Jaw Drop
4	Brow Lowerer	11	Nasolabial Deepener	18	Lip Pucker	27	Mouth Stretch
5	Upper Lid Raiser	12	Lip Corner Puller	20	Lip Stretcher	38	Nostril Dilator
6	Cheek Raiser	14	Dimpler	23	Lip Tightener	39	Nostril Compressor
7	Lid Tightenener	15	Lip Corner Depressor	24	Lip Pressor	43	Eyes Closed

TABLE II
ADOPTED ACTION UNITS IN OUR EXPERIMENTS TO BUILD HYPEREDGES.

Method	Accuracy
AUDN [28]	93.7%
Zero-bias CNN + AD [8]	96.4% $\pm$ 3.1%
D-CNN + SVM	96.5% $\pm$ 2.5%
D-CNN + M-way Softmax	95.7% $\pm$ 1.9%
D-CNN + Hypergraph	**98.6% $\pm$ 2.3%**

TABLE III
PERFORMANCE COMPARISON ON CK+.

Method	Accuracy
DLSVM [9]	71.2%
D-CNN + SVM	73.5%
D-CNN + Softmax	73.7%
D-CNN + Hypergraph	**77.4%**

TABLE IV
PERFORMANCE COMPARISON ON FER2013. DLSVM [9] IS THE WINNING METHOD OF ICML 2013 CHALLENGE FOR REPRESENTATION LEARNING.

demonstrated by extensive experimentation on two popular databases.

REFERENCES

[1] Ying-li Tian, Takeo Kanade, and Jeffrey F. Cohn, "Recognizing action units for facial expression analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 97–115, Feb. 2001.

[2] Yan Tong, Wenhui Liao, and Qiang Ji, "Facial action unit recognition by exploiting their dynamic and semantic relationships," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 10, pp. 1683–1699, 2007.

[3] P. Ekman and W. Friesen, "Facial action coding system: A technique for the measurement of facial movement," *Consulting Psychologists Press*, 1978.

[4] Marian Stewart Bartlett, Gwen Littlewort, Mark Frank, Claudia Lainscsek, Ian Fasel, and Javier Movellan, "Recognizing facial expression: Machine learning and application to spontaneous behavior," .

[5] Jacob Whitehill and Christian W. Omlin, "Haar features for facs au recognition.," in *FG*. 2006, pp. 97–101, IEEE Computer Society.

[6] Guoying Zhao and Matti Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 915–928, 2007.

[7] Amogh Gudi, H. Emrah Tasli, Tim M. den Uyl, and Andreas Maroulis, "Deep learning based FACS action unit occurrence and intensity estimation," in *11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, FG 2015, Ljubljana, Slovenia, May 4-8, 2015*, 2015.

[8] Pooya Khorrami, Tom Le Paine, and Thomas S. Huang, "Do deep neural networks learn facial action units when doing expression recognition?," *CoRR*, vol. abs/1510.02969, 2015.

[9] Yichuan Tang, "Deep learning with linear support vector machines," *Workshop on Representational Learning, ICML*, 2013.

[10] Dengyong Zhou, Jiayuan Huang, and Bernhard Schökopf, "Learning with hypergraphs: Clustering, classification, and embedding," in *NIPS'06*.

[11] Sameer Agarwal, Kristin Branson, and Serge Belongie, "Higher order learning with graphs," in *ICML '06*.

[12] S. Agarwal, J.W. Lim, L. Zelnik Manor, P. Perona, D.J. Kriegman, and S. Belongie, "Beyond pairwise clustering," in *CVPR'05*.

[13] Liang Sun, Shuiwang Ji, and Jieping Ye, "Hypergraph spectral learning for multi-label classification," in *SIG KDD '08*.

[14] Y.C. Huang, Q.S. Liu, and D.N. Metaxas, "Video object segmentation by hypergraph cut," CVPR'09.

[15] Ze Tian, Taehyun Hwang, and Rui Kuang, "A hypergraph-based learning algorithm for classifying gene expression and array cgh data with prior knowledge," *Bioinformatics*, July 2009.

[16] David Lowe, "Object recognition from local scale-invariant features," in *ICCV'99*.

[17] Yuchi Huang, Qingshan Liu, Shaoting Zhang, and Dimitris N. Metaxas, "Image retrieval via probabilistic hypergraph ranking.," in *CVPR*. 2010, pp. 3376–3383, IEEE Computer Society.

[18] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[19] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, 2012.

[20] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.

[21] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson, "Cnn features off-the-shelf: An astounding baseline for recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2014.

[22] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[23] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kande dataset (ck+): A complete facial expression dataset for action unit and emotion-specified expression," in *CVPR4HB*, 2010.

[24] Pierre-Luc Carrier and Aaron Courville, "Facial expression recognition dataset," in *ICML 2013 Challenges in Representation Learning*, 2013.

[25] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schökopf, "Learning with local and global consistency," in *NIPS'03*.

[26] Mengyi Liu, Shaoxin Li, Shiguang Shan, and Xilin Chen, "Au-aware deep networks for facial expression recognition," in *10th IEEE International Conf. on Automatic Face and Gesture Recognition, FG*, 2013.

[27] Tal Hassner, Shai Harel, Eran Paz, and Roee Enbar, "Effective face frontalization in unconstrained images," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

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