Micro-expression Recognition with Small Sample Size by Transferring Long-term Convolutional Neural Network

Su-Jing Wang[a,h], Bing-Jun Li[b], Yong-Jin Liu[b,*], Wen-Jing Yan[c],
Xinyu Ou[d], Xiaohua Huang[e], Feng Xu[f], Xiaolan Fu[g,h]

[a] *CAS Key Laboratory of Behavioral Science, Institute of Psychology, Beijing, 100101, China*

[b] *Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing, P. R. China*

[c] *College of Information Science and Engineering, Northeastern University, Shenyang, China*

[d] *Cadres Online Learning Institute of Yunnan Province, Yunnan Open University, Kunming, 650223, China*

[e] *enter for Machine Vision and Signal Analysis, Faulty of Information Technology and Electrical Engineering, University of Oulu, P. O. Box 4500, FI-90014, Finland*

[f] *Shanghai Key Laboratory of Intelligent Information Processing, Key Laboratory for Information Science of Electromagnetic Waves (MoE), and the School of Computer Science, Fudan University, Shanghai, 200433, China*

[g] *State Key Laboratory of Brain and Cognitive Science, Institute of Psychology, Chinese Academy of Sciences, Beijing, 100101, China*

[h] *Department of Psychology, University of the Chinese Academy of Sciences, Beijing, 100049, China*

Abstract

Micro-expression is one of important clues for detecting lies. Its most outstanding characteristics include short duration and low intensity of movement. Therefore, video clips of high spatial-temporal resolution are much more desired than still images to provide sufficient details. On the other hand, owing to the difficulties to collect and encode micro-expression data, it is small sample size. In this paper, we use only 560 micro-expression video clips to evaluate the proposed network model: Transferring Long-term Convolutional Neural Network (TLCNN). TLCNN uses Deep CNN to extract features from each frame of micro-expression video clips, then feeds them to Long Short

[*] Corresponding author
Email address: liuyongjin@tsinghua.edu.cn (Yong-Jin Liu)

Term Memory (LSTM) which learn the temporal sequence information of micro-expression. Due to the small sample size of micro-expression data, TLCNN uses two steps of transfer learning: (1) transferring from expression data and (2) transferring from single frame of micro-expression video clips, which can be regarded as “big data”. Evaluation on 560 micro-expression video clips collected from three spontaneous databases is performed. The results show that the proposed TLCNN is better than some state-of-the-art algorithms.

Keywords: Micro-expression, Deep Learning, Transferring Learning , Convolutional Neural Network.

1. INTRODUCTION

Lie is an integral and inevitable existence in society that occurs every day and several times within a day [1][2]. The consequences of telling lies (the failed identification of concealed and falsified information) are enormous in many contexts, including suspect interrogations, customs agencies, airport security, and the courtroom [3]. Effective lie detection may help inhibit and avoid potential dangers and harms. The polygraph, a widely used method to detect lies, is invasive because it must be connected to the individual’s body throughout the session [4], where individuals are aware that they are being monitored and may develop countermeasures. Lie detection based on nonverbal cues is unobtrusive, and the individuals being observed are less likely to develop countermeasures.

In USA, officers were trained to judge the potentially dangerous people by their nonverbal behaviors [5]. Among the various nonverbal behaviors, micro-expression is considered as a promising one, because it leaks people’s concealed emotions and may reveal their intent, thus is applicable to detecting lies [6]. In comparison with those connecting apparatus such as polygraph, lie detection based on micro-expressions, which can be captured with hidden camera, is unobtrusive.

Although micro-expression now is gaining more attention and has potential application in a variety of fields, humans have difficulty in detecting and recognizing them. This difficulty results from their short duration, low intensity, and fragmental action units [7][8]. Although there is a debate regarding their duration, the generally accepted limit is 0.5 seconds [8][9]. Micro-expressions are usually very subtle because individuals try to control and repress them. In addition, micro-expressions usually exhibit only parts of the action units of fully-stretched facial expressions.

The automatic recognizing micro-expressions from on-line camera or off-line videos in interrogation interview context may greatly help security officers in detecting the suspects' usual or even deception clues. Therefore, computer vision techniques have the potential to be used in rapid security screening without the need for skilled staff or physical contact.

Research on facial expressions originate from Darwin *et al.* [10]. A previous study conducted by Mehrabian *et al.* [11] has revealed that 55% of messages regarding feelings and attitudes are conveyed via facial expressions. Micro-expression was firstly discovered by Haggard *et al.* [12], which were called rapid expressions that showing repressed emotions at that time. Ekman *et al.* [7] founded this kind of expressions from an inpatient with psychotic who wanted to commit suicide and concealed the negative expression within 1/12 seconds in smiles, and they named it micro-expression. Facial Action Coding System (FACS) [13] and Micro Expression Train Tool (METT) were developed afterwards. Micro-expressions can reveal our authentic emotions, and it is considered to be one of the most important non-verbal leakages and clues (e.g., judging whether someone is lying or honest [6] [14], Clinical Medicine [15] [16] [17], Political Psychology [18]). There are a multitude of researchs concerning facial expressions, however, more knowledge needs to be further studied respecting micro-expressions.

Some studies on micro-expression recognitions have been published in recent years. Polikovsky *et al.* [19] recognized micro-expressions based on 3D-Gradients orientation histogram descriptor. Pfister *et al.* [20] used a Temporal Interpolation Model (TIM) based on Laplacian matrix to normalize the frame numbers of micro-expression video clips. Then, the LBP-TOP [21] was used to extract the motion and appearance features of micro-expressions and multiple kernel learning was used for classification. Wang *et al.* [22] utilized Discriminant Tensor Subspace Analysis (DTSA) which treated a gray micro-expression video clip as a third order tensor and Extreme Learning Machine (ELM) used for classification. Wang *et al.* [23][24] set up a novel color space model, Tensor Independent Color Space (TICS) because color could provide useful information for expression recognition. Then they [25] used the sparse part of Robust PCA (RPCA) [26] to extract the subtle motion information of micro-expression and Local Spatiotemporal Directional Features (LSTD) [27] to extract the local texture features. Yu *et al.* [28] proposed Facial Dynamics Map (FDM) to describe the motion pattern of a micro-expression instance. Liu *et al.* [29] proposed a simple yet effective Main Directional Mean Optical-flow (MDMO) feature for micro-expression recognition. Furthermore, Wang *et al.* [30] proposed a Main Directional Maximal Difference (MDMD) Analysis for micro-expression spotting. Shreve *et al.* [31] used optical strain to spot marco-expression and micro-expression in videos. Huang *et al.* [32] proposed a Spatiotemporal Local Binary Pattern with Integral Projection (STLBP-IP), in which they used integral projection for extracting face shape information and subsequently employed 1-D and 2-D local binary pattern to face shape, for micro-expression recognition. They [33] also proposed Spatiotemporal Local Quantized Pattern (STCLQP), which exploits magnitude and orientation as complementary of sign information, for improving the performance of micro-expression recognition. Patel *et al.* [34] used the pretrained ImageNet-VGG-f CNN to extract the features of each frame of videos and used evolutionary search to select the discriminative feature for micro-expression recognition.

Recently, owing to the rapid development of computer hardware, especially Graphical Processor Unit (GPU), deep learning is applied on many areas such as face recognition [35] and verification [36], and shows outstanding performances. These deep learning methods use multiple processing layers to discover patterns and structures in very large data sets. Each layer learns a concept from the data that subsequent layers build on; the higher the level, the more abstract the concepts that are learned. Deep learning does not depend on prior data processing and automatically extracts features [37]. These advantages and good performances of deep learning are ascribed to big data. However, the number of micro-expression video clips is usually small. Deep learning on data with small sample size may not achieve good performances. To address this problem, we use transfer learning to pre-train a deep convolutional neural network and we propose the Transferring Long-term Convolutional Neural Network (TLCNN) model for micro-expression recognition. In TLCNN, there are two steps of transfer learning: (1) transferring from expression data and (2) transferring from single frame of micro-expression video clips, which can be regarded as "big data". To fully use the temporal information in micro-expression videos, TLCNN also uses Long Short Term Memory (LSTM) to extract temporal features of micro-expression videos from mid-level image representation for each frame images.

The rest of this paper is organized as follows: in Section 2, we briefly review the deep convolutional neural network and the recurrent neural network. In Section 3, we analyze the small sample size problem of micro-expression data and propose Transferring Long-term Convolutional Neural Network (TLCNN) model. Section 4 presents the evaluation on 560 micro-expression video clips collected from three spontaneous databases and the results show that the proposed TLCNN is better than state-of-the-art algorithms. Finally, conclusions are presented in Section 5 and several issues for future work are discussed.

2. Background

2.1. Deep Convolutional Neural Network

A convolutional neural network (CNN) is a special type of a general feed-forward neural network which is specifically designed to deal with still images [38]. Generally, a Deep CNN consists of multiple convolutional layers and pooling layers followed by a few fully-connected layers. In the convolutional layer, the convolution operation is used to extract features from local neighborhood on feature maps in the previous layer. Then an additive bias is applied and the result is passed through an activation function. The notation v_{ij}^{xy} means the value of an unit at position (x, y) in the jth feature map in the ith layer. Then

$$v_{ij}^{xy} = f\left(b_{ij} + \sum_{k}\sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1} w_{ijk}^{pq} v_{(i-1)k}^{(x+p)(y+q)}\right) \tag{1}$$

where $f(\cdot)$ is an activation function, b_{ij} is the bias for this feature map, k is the index over the set of feature maps in the $(i-1)$th layer connected to the current feature map, the kernel weight w_{ijk}^{pq} is the value at the position (p, q) of the kernel connected to the kth feature map, and P_i and Q_i are the height and width of the kernel, respectively. Here, the kernel weight w_{ijk}^{pq} is a special type of weight, which is gotten by learning. In the pooling layers, the resolution of the feature maps is reduced by pooling over local neighborhood on the feature maps in the previous layer.

The CNN described above is called 2D-CNN, because it only extract 2D features from the spatial dimensions. To analyze the video temporal information, Ji *et al.* [39] proposed 3D-CNN. 3D-CNN performs 3D convolutions in the convolution stages of CNNs to extract features from both spatial and temporal dimensions. The 3D convolution is achieved by convolving a 3D kernel to the cube formed by stacking multiple contiguous frames together. By this construction, the feature maps in the convolution layer are connected to multiple contiguous frames in the previous layer, thereby capturing motion information. Formally, the value at position (x, y, z) on the jth feature map in the ith layer is given by

$$v_{ij}^{xyz} = f\left(b_{ij} + \sum_k \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} w_{ijk}^{pqr} v_{(i-1)k}^{(x+p)(y+q)(z+r)}\right) \tag{2}$$

where R_i is the size of the 3D kernel along the temporal dimension, w_{ijk}^{pqr} is the (p, q, r)th value of the kernel connected to the kth feature map in the previous layer. Fig. 1 shows a comparison of 2D and 3D convolutions.

2.2. Recurrent Neural Network

Besides 3D-CNN, recurrent neural network (RNN) can also deal with the video temporal information. RNN is a neural network dealing with an input sequence x_t $(t = 1, 2, \ldots, T)$ and output a corresponding sequence y_t, using an internal hidden state h_t. The RNN sequentially reads each symbol x_t of the input sequence and updates its internal hidden state $\mathbf{h}_t$ according to

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{3}$$

where $f(\cdot)$ is an activation function, W_{xh} is a weight from x_t to h_t and W_{hh} is a weight from h_{t-1} to h_t. RNN can output a prediction y_t at each time step t

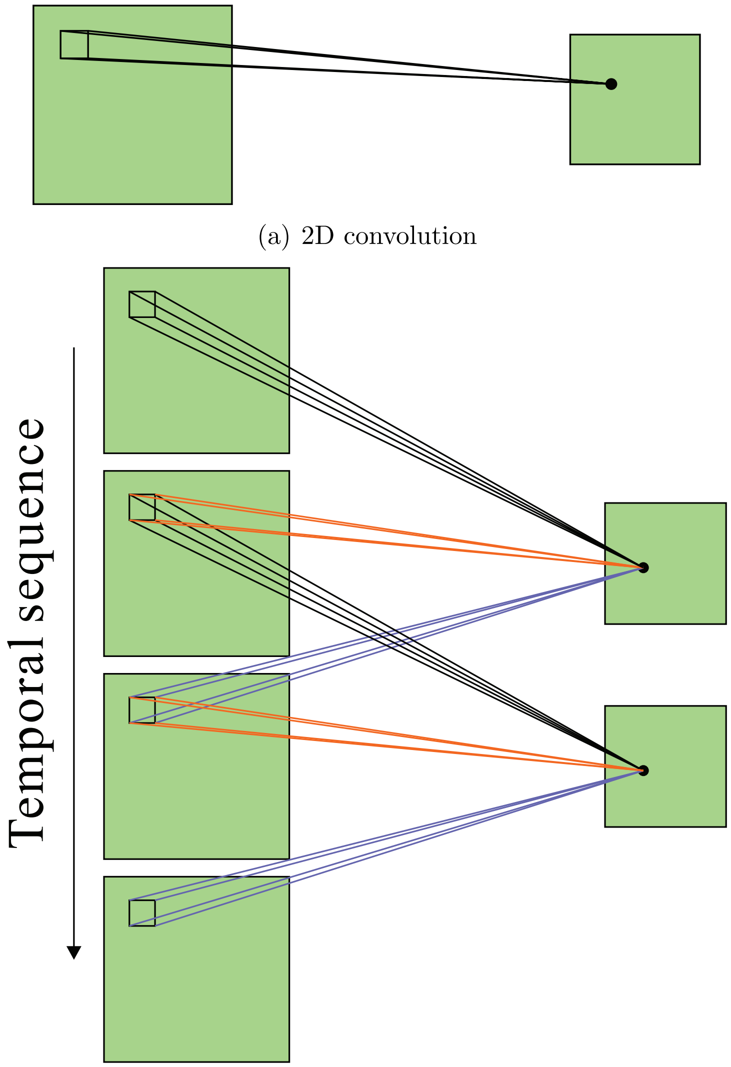
$$y_t = f(W_{hy}h_t + b_y) \tag{4}$$

where W_{hy} is a weight from h_t to y_t.

To solve it, RNN is unfolded the network along the input sequence. Fig. 2 shows part of an unfolded RNN. The unfolded RNN can be viewed as a general neural network to define forward and backward operations.

However, RNN is difficult to be trained to learn long sequences, likely due in part to the vanishing and exploding gradients problem that can result from propagating the gradients down through the many layers of the recurrent network, each corresponding to a particular step t [40]. To address this problem, Long Short Term Memory (LSTM) was proposed. It incorporate memory units to make the network to learn by *Forget Gate* f_t and *Input Gate* i_t when to forget previous hidden states and when to update hidden states given new information.

A LSTM unit is described Fig. 3. Let $\sigma(x) = (1 + e^{-x})^{-1}$ be the sigmoid nonlinearity which squashes real-valued inputs to a $[0, 1]$ range, and let



(a) 2D convolution

(b) 3D convolution

Figure 1: Comparison of (a) 2D and (b) 3D convolutions. In (b) the size of the convolution kernel in the temporal dimension is 3 and the sets of connections are color-coded so that the shared weights are in the same color. In 3D convolution, the same 3D kernel is applied to overlapping 3D cubes in the input video to extract motion features. [39]

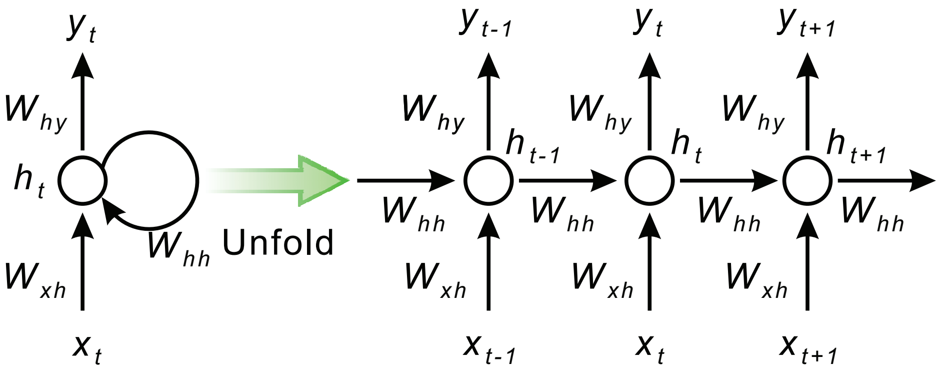


Figure 2: An example of unfolding CNN.

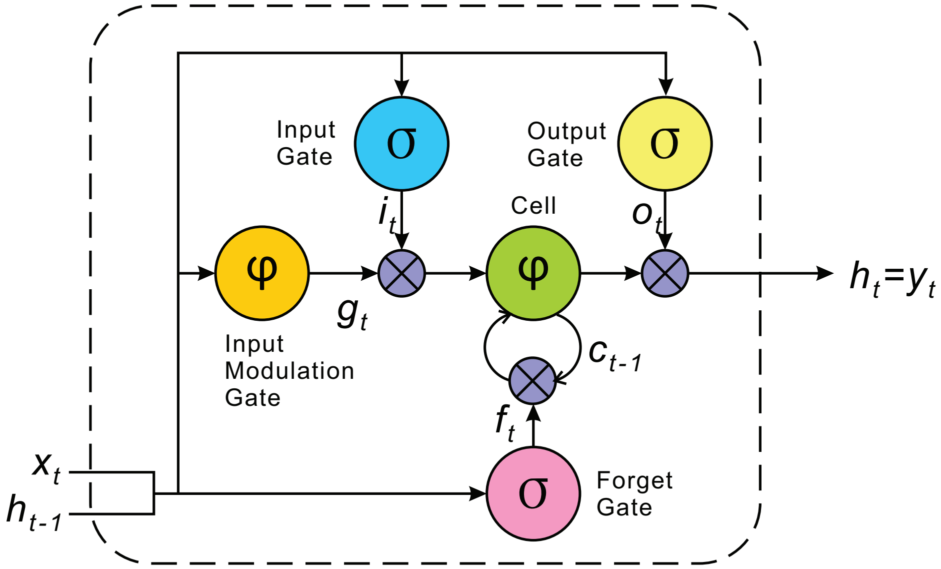


Figure 3: An example of a LSTM unit.

$\phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = 2\sigma(2x) - 1$ be the hyperbolic tangent nonlinearity, similarly squashing its inputs to a $[-1, 1]$ range. The LSTM updates for time t given inputs x_t, h_{t-1}, and c_{t-1} are:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{5}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \tag{6}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{7}$$

$$g_t = \phi(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{8}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{9}$$

$$h_t = o_t \odot \phi(c_t) \tag{10}$$

where o_t is a *Output Gate*, g_t is a *Input Modulation Gate* and c_t is a *memory cell.* The memory cell unit c_t is a summation of two items: the previous memory cell unit c_{t-1} which is modulated by f_t, and g_t, a function of the current input and previous hidden state, modulated by the input gate it. Because i_t and f_t are sigmoidal, their values lie within the range $[0, 1]$, and i_t and f_t can be thought of as knobs that the LSTM learns to selectively forget its previous memory or consider its current input. Likewise, the output gate o_t learns how much of the memory cell to transfer to the hidden state. These additional cells enable the LSTM to learn extremely complex and long-term temporal dynamics the RNN is not capable of learning [40].

3. Transferring Long-term Convolutional Neural Network Model

3.1. Deep Leaning and Big Data

The increase of large-scale data in computational resources make the use of more powerful statistical models become a reality. At the aspect of leveraging large-scale data, deep neural networks have shown superior scaling properties than traditional machine learning methods like Subspace Analysis.

The deep and large networks have shown remarkable results once: (1) large amount of training data has been applied and (2) large scale parallel computing has become available with the development of CPU cores [41] and GPU [42]. Most obviously, it has been confirmed by Krizhevsky et al. [42] that with the use of standard backpropagation, great recognition accuracy could be obtained on a large dataset by very large and deep convolutional neural networks.

For example, DeepFace [35] used 4.4 million labeled faces as training samples to obtain 97.35% accuracy for the face recognition task on LFW face database. DeepID [36] used more 87 thousands training samples to obtain 97.45% for the face verification task on LFW face database.

3.2. Small Sample Size on Micro-expression

However, the number of samples of micro-expression data is very small. Up to now, there are 3 spontaneous micro-expression databases: SMIC [43], CASME [44] and CASME 2 [45]. SMIC contains 164 micro-expression clips induced by 16 participants. Clips from all participants were recorded with a high speed 100fps camera . CASME database contains 195 spontaneous micro-expressions (selecting from 1500 elicited facial movements) filmed under 60 fps. These samples were coded so that the onset, peak and offset frames were tagged. Another micro-expression database, CASME 2, is later developed and contains 247 micro-expression samples from 26 participants. They are selected from nearly 3,000 elicited facial movements. Table 1 lists the above three public micro-expression databases.

Table 1: The existing spontaneous micro-expression databases.

Databasme	Sample Size	Emotion Class	Frames per second	Label
SMIC	164	3	100	Emotion
CASME	195	7	60	Emotion/AUs
CASME 2	247	5	200	Emotion/AUs

Though researchers have great demands on spontaneous micro-expression databases, only very few were developed. The main difficulties lie in micro-expression collecting and annotating. Micro-expression usually occurs when the individual has strong emotions while tries to conceal. Since it's difficult to set a high-stakes situation in a lab, the convenient way to elicit micro-expressions is presenting emotional video clips and asking participants to suppress any facial expressions. This method was adopted to elicit and collect micro-expressions in these 3 databases though had some drawbacks such as that not all participants show (leak) micro-expressions and some very few. In micro-expression annotation, coders takes considerable time and effort to code the duration and AUs. Micro-expressions are currently recognized and defined by its duration [8]; to calculate the duration of facial expressions, researchers have to manually count the frames and ensure it falls within the range of 0.5s in order to classify a facial expression as a micro-expression. Spotting the beginning (onset) and ending (offset) frames of these subtle facial movements is time-consuming and requires intensive manual labor. Moreover, manual coding with FACS is laborious, time-consuming, and strenuous especially for subtle facial movements. Previous studies have indicated that coding, comprehensively, a one minute video footage typically takes over two hours [46]. For very subtle facial expressions (a facial expression with intensity lower than the lowest intensity level depicted in the FACS manual) manual coding is even more demanding and time-consuming. Therefore, annotating is other challenge to develop micro-expression databases.

However, these small samples have "big data". Very high dimensional data are generated from a high-speed and high-resolution camera. A micro-expression video sequence of 0.5 s, filmed at 200 fps, with a resolution of 800×600 would generate a data file of roughly 137 MB. The "big data" is used to pre-train our network.

3.3. Transferring Learning From Expression to Micro-expression

Transfer learning aims to transfer knowledge between related source and target domains [47]. A domain $\mathcal{D}$ consists of two components: a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$, where $X = \{x_1, \ldots, x_n\} \in \mathcal{X}$. Given a specific domain, $\mathcal{D} = \{\mathcal{D}, P(X)\}$, a task consists of two components: a label space $\mathcal{Y}$ and an objective predictive function $f(\cdot)$ (denoted by $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$), which is not observed but can be learned from the training data, which consist of pairs $\{x_i, y_i\}$, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. The function $f(\cdot)$ can be used to predict the corresponding label, $f(x)$, of a new instance x.

Given a source domain $\mathcal{D}_S$ and a domain task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and a target task $\mathcal{T}_T$, *transfer learning* aims to help improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$. We denote the number of elements of $\mathcal{D}_S$ and $\mathcal{D}_T$ as n_S and n_T, respectively. In most cases, $0 \leq n_T \ll n_S$.

In some situations, when the source domain and target domain are not related to each other, brute-force transfer cannot be successful. In the worst case, it may even hurt the performance of learning in the target domain. Here, the source domain is the expression data, and the target domain is the micro-expression data. The source and target domains are different but related, because micro-expression is a special type of expression. Transfer Learning transfers some knowledge, which may be common between different domains, to help improve performance for the target domain or task. Expressions and micro-expressions share the common knowledge, which are similar AUs when expressing emotions, thus have similar texture information. As dynamic facial movements, they have the same temporal pattern, onset phase, apex phase and offset phase. The similar texture and temporal pattern make possible for transfer learning from expressions to micro-expression though the duration of micro-expression is shorter and the intensity usually less.

Not only the source domain and the target domain but also the source task and the target task are related. In our experiments, the source label space is

$$\begin{aligned}\mathcal{Y}_S = \{&Happy, Angry, Sad, Contempt, Disgust, Neutral,\\ &Fear, Surprise, Afraid\}\end{aligned} \tag{11}$$

and the target label space is

$$\mathcal{Y}_T = \{Positive, Negative, Surprise, Others\}, \tag{12}$$

where $Positive = \{Happy\}$, $Surprise = \{Surprise\}$, and $Negative = \{Afraid, Angry, Disgust, Sad, Fear\}$. For micro-expressions, some facial movements are very subtle and difficult to label with basic emotions. Those facial movements with unclear emotion are classified as $Others$. So, the source task and the target task are related. This further develops the performance of transfer learning.

In the other hand, Deep CNN amounts to estimating millions of parameters and requires a very large number of annotated samples [48]. As stated in Section 3.2, however, the number of samples of micro-expression data is only 606 ($n_T = 606$), which is very small compared to existing large face data and expression data. Deep CNN on such small size data cannot guarantee to have a good performance. Compared to micro-expression data, expression data have larger size. In our experiments, 3383 ($n_S = 3383$) expression samples are used. $n_T \ll n_S$ is hold. Meantime, the larger data size, the better performance of deep learning. Once a deep CNN is trained on a large sample size expression dataset, we can use any intermediate representation, such as the feature map from any convolutional layer or the vector representation from any subsequent fully-connected layers, of the whole network for micro-expression. It has been observed that the use of these intermediate representation from the deep CNN as an image descriptor significantly boosts subsequent tasks.

3.4. CNN Architecture

We train our CNN to extract the features from each frame in micro-expression video clips. The overall architecture is shown in Fig. 4 . The CNN net contains five convolutional layers and three fully-connected layers. The output of the last fully-connected layer is fed to a 4-way softmax layer which produces a distribution over the 4 class labels.

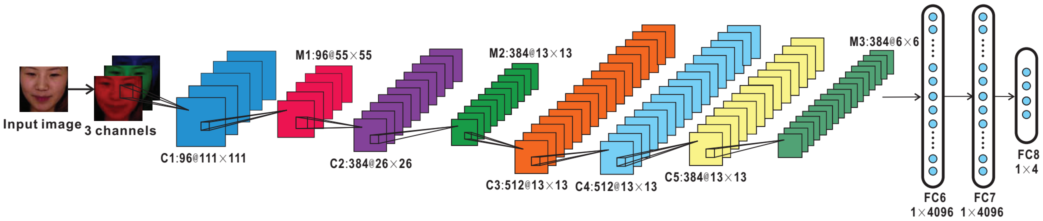


Figure 4: CNN Architecture.

The ReLU non-linearity layer is applied to the output of every convolutional and fully-connected layer. Max-pooling layer follows the first, second and fifth convolutional layer. The response-normalization layer follows the first and second convolutional layer. After the first and second fully-connected layer, there is a dropout layer.

A 3-channels image (RGB) of size 240 × 320 is cropped by the data layer as 227 × 227 × 3 image as the input of the first convolutional layer. The first convolutional layer (C1) filters 227 × 227 × 3 input image with 96 kernels of size 7 × 7 with a stride of 2 pixels, then the 96 feature maps are fed to a max-pooling layer (M1) which takes the max over 3 × 3 spatial neighborhoods with a stride of 2 pixels, separately for each feature map. The second convolutional layer (C2) takes as input the output of the first convolution layer which has a shape of 96 × 55 × 55 and filter it with 384 kernels of size 5 × 5 with a stride of 2 pixels. After activated by the ReLU layer, pooled (M2, parameters same as M1) and response-normalized, the output shape of the second convolutional layer is 384 × 13 × 13. The third (C3), fourth (C4), and fifth (C5) convolutional layer have 512, 512, and 384 kernels separately. All the kernels have the same size of 3 × 3 with pad of 1 pixel and stride of 1 pixel. So the output shape of the third, fourth, and fifth convolutional layer are 512 × 13 × 13, 512 × 13 × 13, and 384 × 13 × 13.

The first five convolutional layers are used to extract low-level features, like texture and simple edges. The following max-pooling layers make the output of the convolutional layers more robust to small registration errors, especially when applied to expression images. Another advantage of the max-pooling layer is the layer parameters can be reduced by half for computation. However, if we use many max-pooling layers, some small and precise features such as detailed facial structures will lose. These features are important for micro-expression recognition. Hence, we only apply the max-pooling layers to the first, second, and fifth convolutional layer. Fig. 5 shows the feature map of C5.

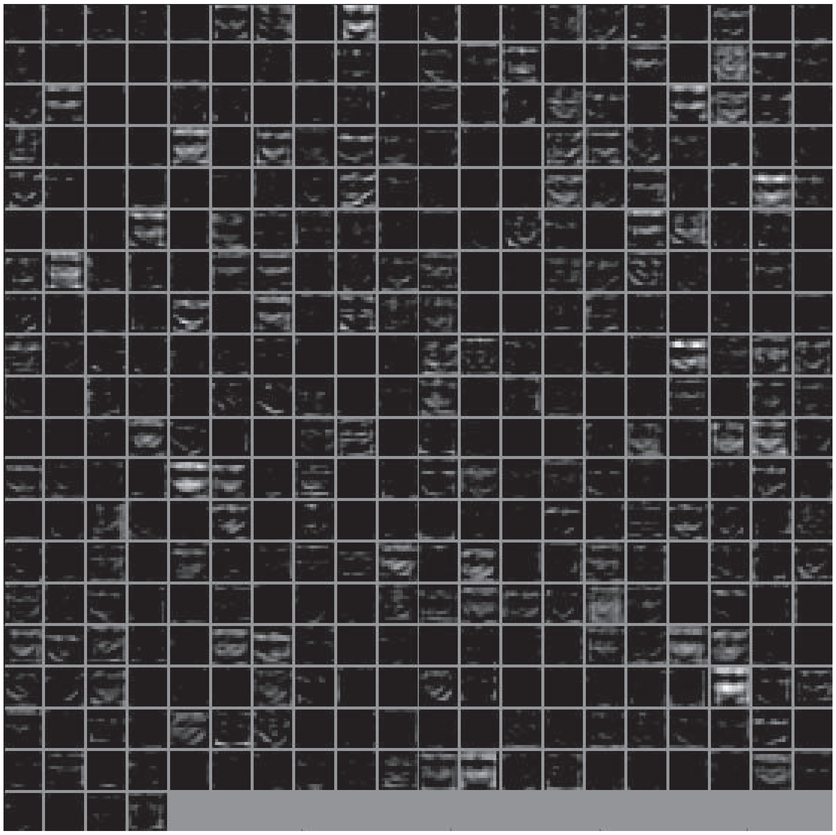


Figure 5: The feature map of C5.

The subsequent layers are fully-connected (FC6, FC7, FC8). The first and second fully-connected layers have 4096 neurons each with dropout ratio of 0.5. The last fully-connected layer has 4 neurons as the classes of micro-expression. A micro-expression has diverse appearance between different areas of facial images. For example, features extracted from areas between eyes and eyebrows show much higher discrimination ability compared to areas between the nose and mouth. So we choose dropout layers to learn combinations of different feature maps from previous layers as various high-level features. Furthermore, the use of dropout layers also can improve the generalized ability of neural networks, prevent over-fitting, and achieve a good performance on the sparse matrix activated by the ReLU function. The fully-connected layers are able to capture correlations between features in distant areas of images such as mouth and eyes etc.

Then the output vector of FC8 layer is fed to a 4-classes softmax layer to produce $p_k = \exp(f(x_k))/\sum_h \exp(f(x_h))$, where x_k is a given input. Then we use the loss function and stochastic gradient backpropagation (SGD) to update the network parameters to optimize the network.

3.5. Transferring Long-term Convolutional Neural Network

We use large sample size expression data to pre-train the above Deep CNN. The trained network includes some expression information which are shared with micro-expression. These information will be transferred to train a network for micro-expression recognition and accelerate the network training speed.

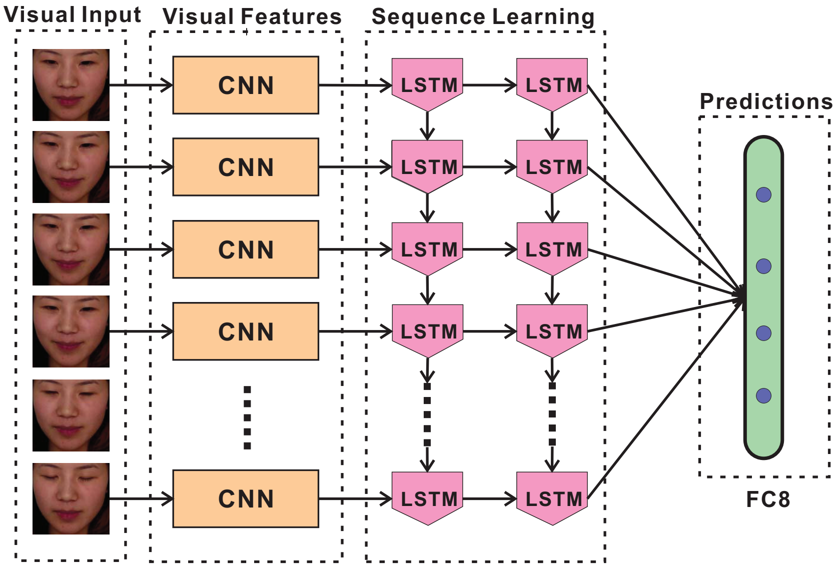


Figure 6: The idea of our method.

The number of micro-expression video clips is small. However, there are dozens or hundreds of frames in each video clip. If each frame is regarded as a sample, micro-expression data will be "big data". We use such "big data" to train again the trained network by the expression data. In this transfer learning step, the source domain is the frame images of micro-expression video and the target domain is the micro-expression video clips. Formally, the source sample space $X_S = \{x_{11}, x_{12}, \ldots, x_{1f}, \ldots, x_{n1}, x_{n2}, \ldots, x_{nf}\}$ and the target sample space $X_T = \{x_1, x_2, \ldots, x_n\}$, where $x_i = \{x_{n1}, x_{n2}, \ldots, x_{nf}\}$. So, the source domain and the target domain are highly related. Further, the source label space $\mathcal{Y}_S$ and the target label space $\mathcal{Y}_T$ are the same. So, the source task and the target task are also highly related. If each micro-expression video clip has 32 frames, than $n_T \ll n_S = 32 \times n_T$. These guarantee that the transfer learning from single frame to video clips can obtain much better performance.

However, only using Deep CNN will lost the dynamic sequence information of micro-expression. To address the problem, we combine Deep CNN and LSTM which can learn the sequence information. Fig. 6 depicts the idea of our method. In the proposed method, Deep CNN extracts features from each frame in micro-expression video clips and produce a fixed-length feature vector representation $\phi_t \in \mathbb{R}^d$. Here, the feature vector representation is the result of FC6 layers in Deep CNN. After computed the feature vector representation of the micro-expression video clip $< \phi_1, \phi_2, \ldots, \phi_T >$, the sequence model then takes over.

Each feature vector representation ϕ_t is regarded as an input x_t of LSTM, which maps an input x_t and a previous timestep hidden state h_{t-1} to an output z_t and updated hidden state h_t. Therefore, inference must be run sequentially, by computing in the following: $h_1 = f_W(x_1, h_0) = f_W(x_1, 0)$, then $h_2 = f_W(x_2, h_1)$, and so on, up to h_T. Finally, all z_t are mapped to labels of micro-expressions by a full-connected layer (FC8). Finally, we train jointly Deep CNN and LSTM together to improve the final accuracy.

4. EXPERIMENTS

4.1. Pre-train Network by Expression Data

The data we use to pre-train our Deep CNN model is facial expression data. The standard that we choose expression database is based on six basic emotions. Besides, the direction of face and the intensity of expression is also taken into consideration. According to these, we strictly selected part of expression samples from four facial expression databases, Karolinska Directed Emotional Faces [49], MMI Facial Expression Database [50], Radboud Faces

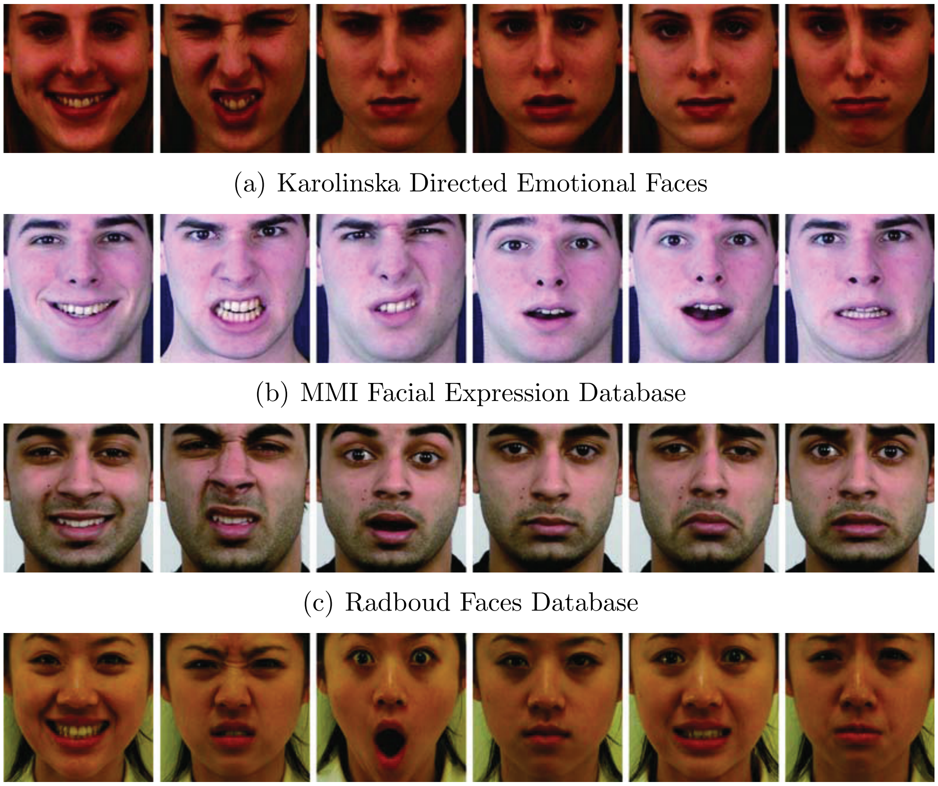
Database [51], Taiwanese Facial Expression Image Database [52]. Karolinska Directed Emotional Faces (KDEF) is a set of totally 4900 pictures of human facial expressions of emotion. It was originally developed to be used for psychological and medical research purpose. The dataset contains 70 subjects, displaying 7 different facial emotions from 5 different angles each. MMI Facial Expression Database(MMI) consists of over 2900 videos and high-resolution still images of 75 subjects, most of the videos and images are annotated for the presence of AUs and emotion. Radboud Faces Database (Radboud) consists of 67 subjects including adults and children, each have 8 categories of emotions from 3 gaze directions of left, frontal and right. Taiwanese Facial Expression Image Database (TFEID) consists of 7200 stimuli captured from 40 subjects (20 males), each with 8 facial expressions: neutral, anger, contempt, disgust, fear, happiness, sadness and surprise. Subjects were asked to gaze at two different angles (0° and 45°). Each expression includes two kinds of intensities (high and slight). Table 2 lists the four expression databases. Fig. 7 shows the cropped samples in the four databases.

Table 2: Four expression databases.

Database	Sample Size	Emotion Class	Emotion
KDEF	980	7	afraid, angry, disgust, happy, neutral, sad, surprise
MMI	176	6	neutral, anger, disgust, fear, happy, sad
Radboud	1608	8	happy, angry, sad, contempt, disgust, neutral, fear, surprise
TFEID	619	8	neutral, anger, contempt, disgust, fear, happy, sad, surprise

We select 980, 176, 1608, 619 images separately from the four databases. Totally, we use 3383 images to pre-train a Deep CNN model and categorize them into 9 classes: neutral, anger, disgust, fear, happy, sad, surprise, shame, contempt. Before feeding these images to the network, face areas are cropped from original images. We use 66 facial feature points detected by Discriminative Response Map Fitting (DRMF) [53] method to locate facial areas in images. These 66 facial feature points lead to good performance on describing facial features such as eyebrow, ear, jaw, lips and locating facial region. Fig. 8 shows these 66 feature points by DRMF and a cropped face area. So we can obtain facial area in our training by cropping our images with a rectangle that is mostly suitable for containing all the feature points. We split all the images into 5 parts, four parts are used for training, the other part is used for validation.

We use expression data to pre-train the network as described in Section



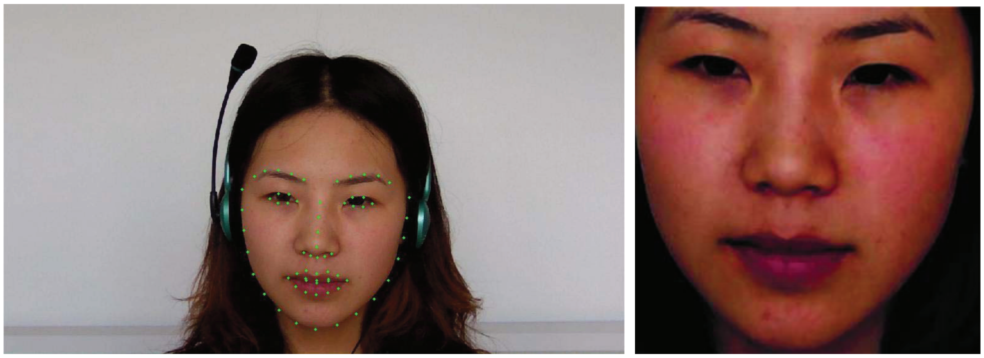
(a) Karolinska Directed Emotional Faces

(b) MMI Facial Expression Database

(c) Radboud Faces Database

(d) Taiwanese Facial Expression Image Database

Figure 7: The examples in the four expression databases.



(a) 66 feature points

(b) cropped face area

Figure 8: An example of face area cropping by using 66 feature points from DRMF.

3.4 which is modified the FC8 layer as 9 neural unit corresponding to 9 expression labels. When the change of test accuracy does not exceed a small threshold in contiguous iterations, the training process is terminated. The final accuracy is 94% (see Fig. 9).

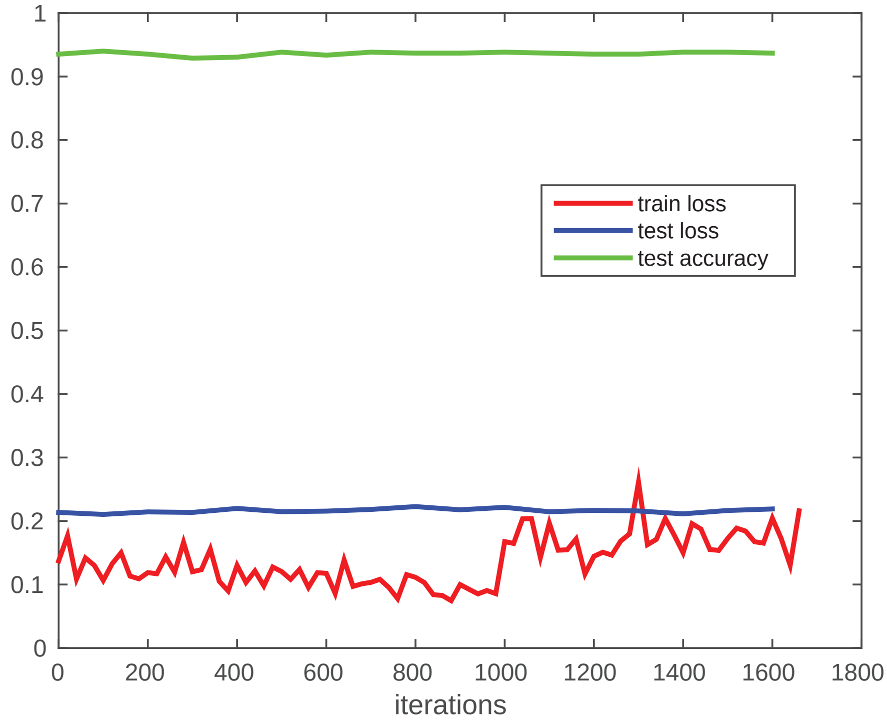


Figure 9: The iteration curves when to pre-train network by expression data.

4.2. Train Network by Micro-expression Data

We use three spontaneous micro-expression databases described in Section 3.2. Among all the samples in the above three micro-expression databases, several samples in which the 66 facial feature points in the first frame could not be detected correctly by the DRMF method were removed. At last, we selected 167, 236, 157 samples separately from above three micro-expression databases. These all 560 samples were categorized into 4 classes: Positive (91 samples), Negative (179 samples), Surprise (75 samples), and Others (214 samples).

Micro-expressions are featured by shorter duration, lower intensity and usually partial movements, which makes the emotion classification difficult since we can't simply label the emotion types as FACS recommends (which is developed based on regular facial expressions). And we find that different databases used different criteria to annotate the micro-expressions, which incurs problems when training the models for recognition. Therefore, we re-classify the micro-expressions into 3 or 4 classes, e.g. positive, negative, and surprise. Positive one refers to happiness, which is easily elicited and have distinct features. Negative ones contain disgust, sadness, fear, et al., which are difficult to distinguish between one another. Surprise, which is not necessary to be positive or negative, has its own distinct pattern of AUs and indicates feelings of unexpectedness. With this re-classification, these micro-expressions can be divided into 3 groups and have clear meaning in a more general level. The "Others" micro-expressions are ambiguous in emotional meaning, even without clear distinguishing between positive and negative. These micro-expressions demonstrate there is something in the persons' mind, but have to be further interpreted according to the situations. Using such classification (4 class), which is feasible from psychological perspective, different databases can be compatible to each other.

All the pictures were cropped with a rectangle generated by 66 facial feature points of the first frame to get facial regions. Considering the difference of frame rate between 3 micro-expressions, we used temporal interpolation model (TIM) to temporally interpolate the sample video clip to normalized frame number of 32 and 64. We split the images into 5 parts, four parts are used for training, the other part is used for testing. There are 448 samples in the training set, and 112 samples in testing set.

For TIM32, each micro-expression video clip has 32 frames. 560 video clips have totally $32 \times 560 = 17920$ frames. Each frame is regarded as a sample. So, there are $32 \times 448 = 14336$ samples in the training set, and $32 \times 112 = 3584$ samples in testing set. These data are used to train the network which is trained by expression data described in Section 4.1. The momentum of SGD is set to 0.9, and batch size is set to 224. We have set an equal learning rate for all trainable layers to 0.001, which is decreased by an order of magnitude every 2560 iterations. At the time of 10000 iterations approximately, we find the validation error become stable. In order to prevent the local optimal solution, we choose the model which had the best accuracy to fine-tune with basic learning rate resetting to 0.0005. Then we got the Deep CNN model with the best accuracy after approximately 10000 iterations.

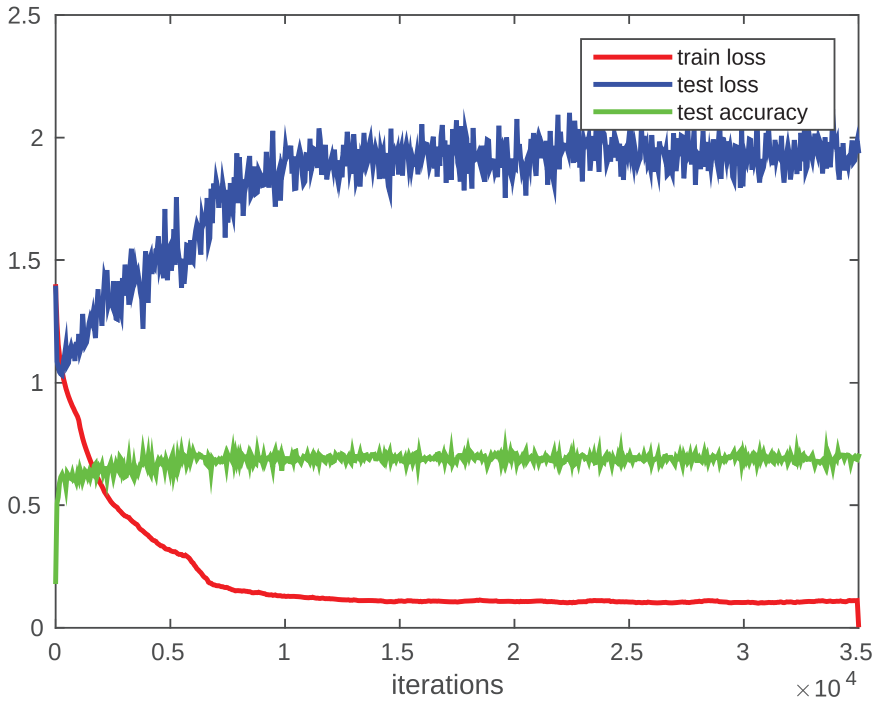


Figure 10: The iteration curves when to jointly train the Deep CNN and LSTM.

In the following, we use the trained Deep CNN to extract features from each frame of micro-expression, then feed them to LSTM and jointly train Deep CNN and LSTM. In order to improve the generalization of the network and data diversity, The consecutive 16 frames are randomly chosen from a micro-expression video. Each batch includes 16 videos. One epoch over the whole data needs 112 batch iterations. The learning rate for all trainable layers are set to 0.01, which is automatically decreased to 10% of the previous learning rate after 5600 iteration. The training is stopped after 20000 iterations and the best accuracy is chosen. Then, same with the previous CNN training, we fine-tune the LSTM from the chosen model. At last, the best accuracy is chosen. Fig. 10 shows the iteration curves.

In order to evaluate the proposed network model for micro-expression recognition, we conduct 3D-CNN, MDMO, FDM, LBP-TOP, STLBP-IP and

STCLQP on the same data.

The net we use to train 3D-CNN is similar to the net we use to train TLCNN, which also has five convolutional layers and three fully-connected layers. The convolutional layers have 64, 128, 256, 256 and 256 neural units separately. The three fully-connected layers have 2048, 2048 and 4 neural units each. The number of neural units of the fully-connected layer is correspond to 4 micro-expression labels. Considering the memory of GPU, the input frame size is 128 × 171. The momentum of SGD is set to 0.9. For TIM32 setup, the basic learning rate is 0.003, which would be automatically decreased to 10% of itself after 20000 iterations, the length of video clips is 32, the batch size is 7. For TIM64 setup, the basic learning rate is 0.001, which would be automatic decreased to 10% of itself after 20000 iterations, the length of video clips is 64, the batch size is 4.

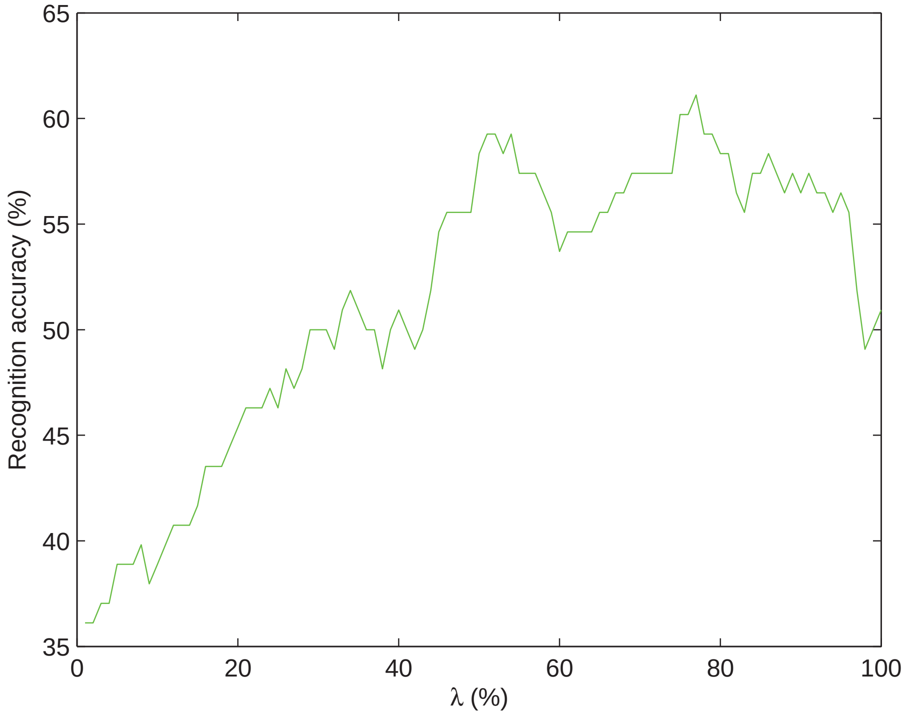


Figure 11: The relation between λ and the recognition accuracy.

MDMO feature could be obtained by the following steps. 1) The facial region is partitioned into 36 ROIs with the 66 facial feature points marked by Discriminative Response Map Fitting (DRMF) [53]. 2) The optical flow $[V_x^i, V_y^i]^T$ between the first frame f_1 and each f_i of subsequent frames is calculated, and is converted into polar coordinates (ρ_i, θ_i), where ρ_i and θ_i are the magnitude and direction of the optical flow vectors respectively. 3) All the optical flow vectors are classified into 8 bins according to their directions. For each ROI, choose the bin in which the number of vectors is maximum from all frames and compute a mean vector to represent this ROI, then we get the 36×2 optical flow feature vector for each sequence; 4) To balance the effect of magnitude and direction, introduce one parameter $\lambda \in (0, 1)$, then the MDMO feature can be represented as $\phi = (\lambda\rho_1, \lambda\rho_2, \ldots, \lambda\rho_{36}, (1 - \lambda)\theta_1, (1 - \lambda)\theta_2, \ldots, (1 - \lambda)\theta_{36})$. We put the MDMO features into the SVM with the polynomial kernel function to classify. Fig. 11 shows the relation between λ and the recognition accuracy. When $\lambda = 0.77$, MDMO obtains the best accuracy 60.11%.

FDM first extracts dense optical flow fields, and then divides the optical flow fields into $P \times Q \times \lfloor \frac{T}{\tau} \rfloor$ (T is the number of frame.) smaller cuboids of size. Each cuboid is small enough such that the motion vectors within it should describe almost identical motion pattern. A sophisticated algorithm is applied to find an optimal direction for each small cuboid. The optical directions are further quantized and linked as the Facial Dynamics Map. The final classification is performed by an SVM with RBF kernel. In our experiment, $P \times Q$ is selected among 4×4, 6×6, 8×8, 12×12, 16×16. For TIM32, τ is selected among 3, 6, 10; and for TIM64, τ is selected among 7, 10, 12. Please note in FDM, we may ignore some optical flow frames under certain parameters. For example, in TIM32, we have 31 optical flow frames. When we set τ to 3, we get 10 batches (31 / 3), and the final optical flow frames is discarded. Therefore, we should avoid discarding too many frames when we select τ. We choose the best accuracy as the final accuracy in each training.

The LBP description from three orthogonal planes (LBP-TOP) [21] is a dynamic texture operator extended from Local Binary Patterns (LBP) and have been successfully applied on expression recognition and micro-expression recognition. In fact, LBP-TOP calculates three LBP codes from three orthogonal planes of 3D objects such as micro-expression video clips (see Fig. 12) and concatenates them into a LBP-TOP code. Here, we calculate three LBP-TOP codes from R, G and B color component channels and concatenate them into a long code which is fed to SVM. The performance

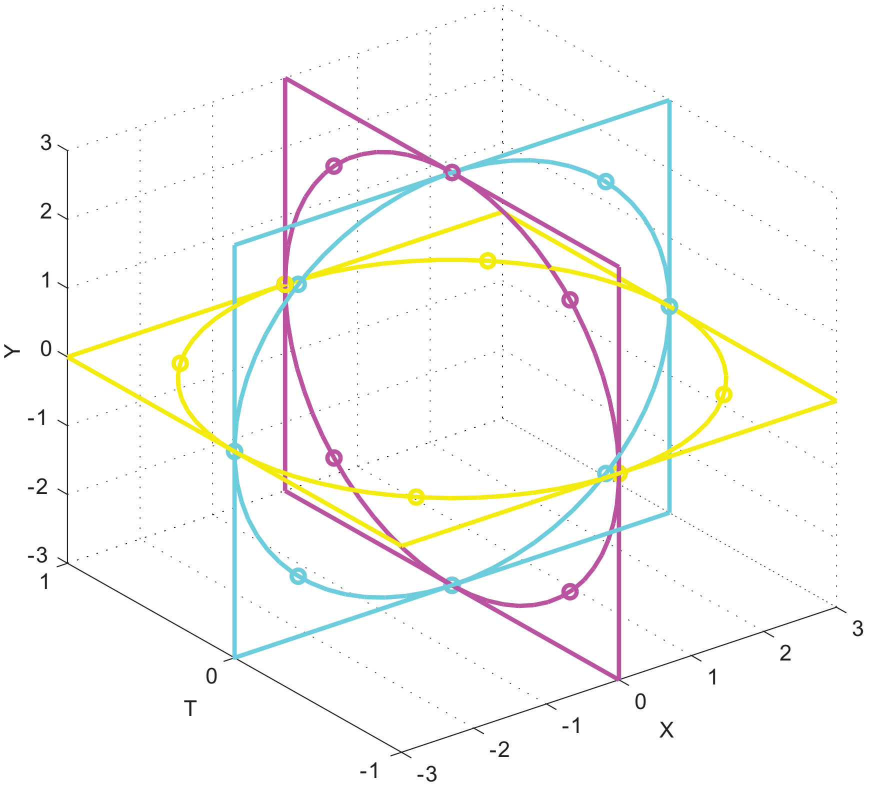


Figure 12: A diagram of LBP-TOP.

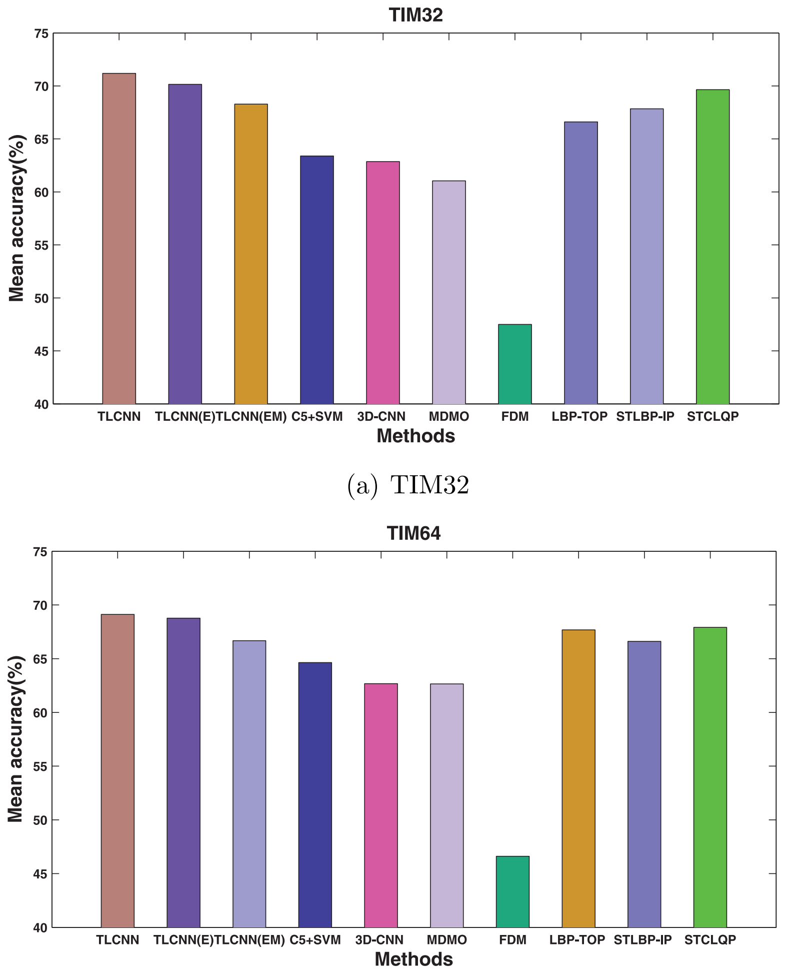
of LBP-TOP on RGB color space is better than that on gray [24]. Because LBP-TOP is a local feature extraction method, facial images are divided into 6×6 patches. For LBP-TOP, the radii in axes X, Y, T were assigned as 3. The number of neighboring points (be marked as P) in the XY, XT and YT planes all were set as 8. The uniform pattern is used in LBP coding.

STLBP-IP, firstly, used integral projection to extract the shape information of all frames in one video. Secondly, for appearance features, it further used 1D Local binary pattern (1DLBP) to obtain their features. Shape information of all frames is constituted as a new temporal texture images. Then 2D Local binary pattern (2DLBP) is utilized to extract texture feature as temporal features. In our implementation, we used STLBP-IP on gray-level video. Following [50], facial images are divided into 8×9 blocks. For 1DLBP, the number of neighboring points (be marked as P) in horizontal and vertical projection were assigned as 8. For 2DLBP, the radii and number of neighboring points in a temporal texture images were set as 3 and P, respectively. The uniform pattern is used in 1DLBP and 2DLBP coding. SVM with Chi-square kernel was employed, in which the penalty parameter was grid-searched.

STCLQP was employed on gray-level video. Facial images are divided into 8×8 blocks, in which the feature of each block is extracted by using STCLQP. For orientation information, Gaussian kernel based on 3×3 pixel sizes is used. The level of orientation estimation and quantization level are set as 16 and 4. For efficient computational cost, we employed one circular neighbor topology $(2, 16)$ with 16 sampling points and radius 2 around the central point. The codebook of size 20 is studied by using k-means method. The dimensionality of sign, magnitude and orientation features was reduced by Supervised Locality Preserving Projection, and reduced features are concatenated into one feature vector. SVM with linear kernel was employed, in which the penalty parameter was grid-searched.

In order to evaluate the transferring effect, we train the network without transferring from the expression data and denote it as *TLCNN(E)*. Furthermore, we only jointly train the Deep CNN and LSTM without transferring from expression information and single frame information and denote it as *TLCNN(EM)*. In addition, we also use the feature map of C5 layer to extract features and feed them into SVM. Table 3 lists the results of these methods. Fig. 13 shows the bar graphs of the mean accuracies of these methods.

Although the different methods obtain the best performances in different folds, the proposed TLCNN obtain the best mean performances both on



(a) TIM32

(b) TIM64

Figure 13: The bar graphs of the mean accuracies of these methods.

Table 3: The results of these methods.

Methods	TIM32						TIM64					
	Fold1	Fold2	Fold3	Fold4	Fold5	Mean	Fold1	Fold2	Fold3	Fold4	Fold5	Mean
TLCNN	**76.17**	**68.95**	67.77	72.32	70.73	**71.19**	**74.47**	64.62	66.10	70.98	69.41	**69.12**
TLCNN(E)	74.67	67.08	66.77	**72.43**	69.81	70.15	73.20	66.88	65.32	**71.12**	67.31	68.77
TLCNN(EM)	71.54	65.63	66.60	70.01	67.69	68.29	69.56	64.72	65.42	66.71	66.98	66.68
C5+SVM	64.29	65.18	64.29	57.14	66.07	63.39	64.29	65.18	66.07	60.71	66.96	64.64
3D-CNN	64.29	64.29	55.36	67.86	62.50	62.86	63.39	63.39	56.25	68.75	61.61	62.68
MDMO	61.11	61.26	61.47	64.15	57.27	61.05	65.74	58.56	66.06	62.26	60.61	62.65
FDM	53.57	41.96	48.21	49.11	44.64	47.50	47.32	50.89	43.75	45.54	45.54	46.61
LBP-TOP	65.18	65.18	66.96	63.39	**72.32**	66.61	62.50	**66.96**	67.86	66.96	**74.11**	67.68
STLBP-IP	66.96	63.29	**73.21**	67.86	67.86	67.84	66.96	62.50	**72.32**	65.18	66.07	66.61
STCLQP	69.64	65.17	72.32	69.64	71.43	69.64	67.86	66.07	70.54	69.64	65.50	67.92

TIM32 and on TIM64. Compared to TLCNN, TLCNN(E), TLCNN(EM), and 3D-CNN (the methods with less transferred information) are lower about 3%. So the transferred information takes effect. In Fold4, TLCNN(E) obtains slily better performances than TLCNN. This is related with the training samples distribution. We also see that the performance of C5+SVM is worse than those of TLCNN, TLCNN(E), and TLCNN(EM). The reason is that C5+SVM doesn't have the temporal sequence information compared to TLCNN.

To prove the effectiveness of TLCNN and transferring learning from expression data and single frames, we also conducted statistical analyses (T-tests) to examine the difference in performance between TLCNN and other algorithms. The performance of TLCNN is statistically better than those of C5+SVM, 3D-CNN, MDMO, and FDM, all $p<0.05$. TLCNN performs marginally significant better than TLCNN(EM), LBP-TOP, STLBP-IP, and STCLQP *(0.05<p<0.1)*. Statistically, we observed no difference in performance between TLCNN and TLCNN(E) all $p>0.2$, noting that TLCNN(E) used a step that transfers learning from single frames to video clips.

In these methods, MDMO and FDM are methods based on optical flow and have the worst performances. TLCNN(EM), C5+SVM and 3D-CNN are deep learning methods without transfer learning and have worse performance than LBP-TOP, STLBP-IP, and STCLQP which are methods based on LBP. TLCNN and TLCNN(E) are deep learning methods with transfer learning and have the best performances.

5. CONCLUSION

In this paper, we proposed Transfer Long-term Convolutional Neural Network (TLCNN), which uses Deep CNN to extract features from each frames of micro-expression video clips, then feed them to Long Short Term Memory (LSTM) which learns the temporal sequence information of micro-expression. For the two outstanding characteristics of micro-expression, TLCNN uses two steps of transfer learning: (1) transferring from expression data and (2) transferring from single frame of micro-expression video clips, which can be regarded as "big data". The experiment results on three spontaneous micro-expression databases show that the proposed TLCNN is better than some state-of-the-art algorithms namely D-CNN, MDMO, FDM, LBP-TOP, STLBP-IP and STCLQP.

Acknowledgments

This paper is supported in part by grants from the National Natural Science Foundation of China (61772511, 61379095, U1736220).

References

[1] B. M. DePaulo, D. A. Kashy, S. E. Kirkendol, M. M. Wyer, J. A. Epstein, Lying in everyday life., Journal of personality and social psychology 70 (1996) 979.

[2] K. B. Serota, T. R. Levine, F. J. Boster, The prevalence of lying in america: Three studies of self-reported lies, Human Communication Research 36 (2010) 2–25.

[3] S. Porter, L. Ten Brinke, Reading between the lies identifying concealed and falsified emotions in universal facial expressions, Psychological Science 19 (2008) 508–514.

[4] S. M. Lajevardi, Z. M. Hussain, Automatic facial expression recognition: feature extraction and selection, Signal, Image and video processing 6 (2012) 159–169.

[5] S. Weinberger, Airport security: intent to deceive?, Nature 465 (2010) 412–415.

[6] P. Ekman, Lie catching and microexpressions, The philosophy of deception (2009) 118–133.

[7] P. Ekman, W. V. Friesen, Nonverbal leakage and clues to deception, Psychiatry 32 (1969) 88–106.

[8] W.-J. Yan, Q. Wu, J. Liang, Y.-H. Chen, X. Fu, How fast are the leaked facial expressions: The duration of micro-expressions, Journal of Nonverbal Behavior 37 (2013) 217–230.

[9] D. Matsumoto, H. S. Hwang, Evidence for training the ability to read microexpressions of emotion, Motivation and Emotion 35 (2011) 181–191.

[10] C. Darwin, The expression of the emotions in man and animals, volume 526, University of Chicago press, 1965.

[11] A. Mehrabian, Communication without words, Psychological today 2 (1968) 53–55.

[12] E. A. Haggard, K. S. Isaacs, Micromomentary facial expressions as indicators of ego mechanisms in psychotherapy, Springer, 1966.

[13] P. Ekman, W. Friesen, Facial action coding system (1977).

[14] G. Warren, E. Schertler, P. Bull, Detecting deception from emotional and unemotional cues, Journal of Nonverbal Behavior 33 (2009) 59–69.

[15] T. A. Russell, E. Chu, M. L. Phillips, A pilot study to investigate the effectiveness of emotion recognition remediation in schizophrenia using the micro-expression training tool, British Journal of Clinical Psychology 45 (2006) 579–583.

[16] T. A. Russell, M. J. Green, I. Simpson, M. Coltheart, Remediation of facial emotion perception in schizophrenia: concomitant changes in visual attention, Schizophrenia research 103 (2008) 248–256.

[17] M. Swart, R. Kortekaas, A. Aleman, Dealing with feelings: characterization of trait alexithymia on emotion regulation strategies and cognitive-emotional processing, PLoS One 4 (2009) e5751.

[18] P. A. Stewart, B. M. Waller, J. N. Schubert, Presidential speechmaking style: Emotional response to micro-expressions of facial affect, Motivation and Emotion 33 (2009) 125–135.

[19] S. Polikovsky, Y. Kameda, Y. Ohta, Facial micro-expressions recognition using high speed camera and 3d-gradient descriptor, in: 3rd International Conference on Imaging for Crime Detection and Prevention (ICDP 2009), 2009.

[20] T. Pfister, X. Li, G. Zhao, M. Pietikäinen, Recognising spontaneous facial micro-expressions, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 1449–1456.

[21] G. Zhao, M. Pietikainen, Dynamic texture recognition using local binary patterns with an application to facial expressions, Pattern Analysis and Machine Intelligence, IEEE Transactions on 29 (2007) 915–928.

[22] S.-J. Wang, H.-L. Chen, W.-J. Yan, Y.-H. Chen, X. Fu, Face recognition and micro-expression recognition based on discriminant tensor subspace analysis plus extreme learning machine, Neural processing letters 39 (2014) 25–43.

[23] S.-J. Wang, W.-J. Yan, X. Li, G. Zhao, X. Fu, Micro-expression recognition using dynamic textures on tensor independent color space, in: Pattern Recognition (ICPR), 2014 22nd International Conference on, IEEE, 2014, pp. 4678–4683.

[24] S.-J. Wang, W.-J. Yan, X. Li, G. Zhao, C.-G. Zhou, X. Fu, M. Yang, J. Tao, Micro-expression recognition using color spaces, IEEE Transactions on Image Processing 24 (2015) 6034–6047.

[25] S.-J. Wang, W.-J. Yan, G. Zhao, X. Fu, C.-G. Zhou, Micro-expression recognition using robust principal component analysis and local spatiotemporal directional features, in: Computer Vision-ECCV 2014 Workshops, Springer, 2014, pp. 325–338.

[26] J. Wright, A. Ganesh, S. Rao, Y. Peng, Y. Ma, Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization, in: Advances in neural information processing systems, 2009, pp. 2080–2088.

[27] G. Zhao, M. Pietikäinen, Visual speaker identification with spatiotemporal directional features, in: Image Analysis and Recognition, Springer, 2013, pp. 1–10.

[28] F. Xu, J. Zhang, J. Wang, Microexpression identification and categorization using a facial dynamics map, Affective Computing, IEEE Transactions on PP (2016) 1–1.

[29] Y.-J. Liu, J.-K. Zhang, W.-J. Yan, S.-J. Wang, G. Zhao, X. Fu, A main directional mean optical flow feature for spontaneous micro-expression recognition, IEEE Transactions on Affective Computing PP (2015) 1–1.

[30] S.-J. Wang, S. Wu, X. Qian, J. Li, X. Fu, A main directional maximal difference analysis for spotting facial movements from long-term videos, Neurocomputing 230 (2017) 382–389.

[31] M. Shreve, J. Brizzi, S. Fefilatyev, T. Luguev, D. Goldgof, S. Sarkar, Automatic expression spotting in videos, Image and Vision Computing 32 (2014) 476–486.

[32] X. Huang, S. J. Wang, G. Zhao, M. Piteikainen, Facial micro-expression recognition using spatiotemporal local binary pattern with integral projection, in: Computer Vision Workshop (ICCVW), 2015 IEEE International Conference on, 2015, pp. 1–9. doi:10.1109/ICCVW.2015.10.

[33] X. Huang, G. Zhao, X. Hong, W. Zheng, M. Pietikinen, Spontaneous facial micro-expression analysis using spatiotemporal completed local quantized patterns, Neurocomputing 175, Part A (2016) 564 – 578.

[34] D. Patel, X. Hong, G. Zhao, Selective deep features for micro-expression recognition, in: 2016 23rd International Conference on Pattern Recognition (ICPR), 2016, pp. 2258–2263. doi:10.1109/ICPR.2016.7899972.

[35] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: Closing the gap to human-level performance in face verification, in: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE, 2014, pp. 1701–1708.

[36] Y. Sun, X. Wang, X. Tang, Deep learning face representation from predicting 10,000 classes, in: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE, 2014, pp. 1891–1898.

[37] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.

[38] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (1998) 2278–2324.

[39] S. Ji, W. Xu, M. Yang, K. Yu, 3d convolutional neural networks for human action recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (2013) 221–231.

[40] 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, arXiv preprint arXiv:1411.4389 (2014).

[41] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le, et al., Large scale distributed deep networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1223–1231.

[42] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[43] X. Li, T. Pfister, X. Huang, G. Zhao, M. Pietikainen, A spontaneous micro-expression database: Inducement, collection and baseline, in: Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on, IEEE, 2013, pp. 1–6.

[44] W.-J. Yan, Q. Wu, Y.-J. Liu, S.-J. Wang, X. Fu, Casme database: a dataset of spontaneous micro-expressions collected from neutralized faces, in: Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on, IEEE, 2013, pp. 1–7.

[45] W.-J. Yan, X. Li, S.-J. Wang, G. Zhao, Y.-J. Liu, Y.-H. Chen, X. Fu, Casme ii: An improved spontaneous micro-expression database and the baseline evaluation, PloS one 9 (2014).

[46] M. Bartlett, G. Littlewort, J. Whitehill, E. Vural, T. Wu, K. Lee, A. Erçil, M. Cetin, J. Movellan, Insights on spontaneous facial expressions from automatic expression measurement, Dynamic Faces: Insights from Experiments and Computation (2010).

[47] S. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on Knowledge and Data Engineering 22 (2010) 1345–1359.

[48] M. Oquab, L. Bottou, I. Laptev, J. Sivic, Learning and transferring mid-level image representations using convolutional neural networks, in: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, 2014, pp. 1717–1724. doi:10.1109/CVPR.2014.222.

[49] D. Lundqvist, A. Flykt, A. Öhman, The karolinska directed emotional faces (kdef), CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet (1998) 91–630.

[50] M. Valstar, M. Pantic, Induced disgust, happiness and surprise: an addition to the mmi facial expression database, in: Proc. 3rd Intern. Workshop on EMOTION (satellite of LREC): Corpora for Research on Emotion and Affect, 2010, p. 65.

[51] O. Langner, R. Dotsch, G. Bijlstra, D. H. Wigboldus, S. T. Hawk, A. van Knippenberg, Presentation and validation of the radboud faces database, Cognition and Emotion 24 (2010) 1377–1388.

[52] L.-F. Chen, Y.-S. Yen, Taiwanese facial expression image database, Brain Mapping Labratory, Institute of Brain Science, National Yang-Ming University, Taipei, Taiwan (2007).

[53] A. Asthana, S. Zafeiriou, S. Cheng, M. Pantic, Robust discriminative response map fitting with constrained local models, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE, 2013, pp. 3444–3451.