Deeply Learning Deformable Facial Action Parts Model for Dynamic Expression Analysis

Mengyi Liu[1,2], Shaoxin Li[1,2], Shiguang Shan[1(✉)], Ruiping Wang[1], and Xilin Chen[1,3]

[1] Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China
{sgshan,wangruiping,xlchen}@ict.ac.cn
[2] University of Chinese Academy of Sciences (UCAS), Beijing 100049, China
[3] Department of Computer Science and Engineering, University of Oulu, Oulu, Finland
{mengyi.liu,shaoxin.li}@vipl.ict.ac.cn

Abstract. Expressions are facial activities invoked by sets of muscle motions, which would give rise to large variations in appearance mainly around facial parts. Therefore, for visual-based expression analysis, localizing the action parts and encoding them effectively become two essential but challenging problems. To take them into account jointly for expression analysis, in this paper, we propose to adapt 3D Convolutional Neural Networks (3D CNN) with deformable action parts constraints. Specifically, we incorporate a deformable parts learning component into the 3D CNN framework, which can detect specific facial action parts under the structured spatial constraints, and obtain the discriminative part-based representation simultaneously. The proposed method is evaluated on two posed expression datasets, CK+, MMI, and a spontaneous dataset FERA. We show that, besides achieving state-of-the-art expression recognition accuracy, our method also enjoys the intuitive appeal that the part detection map can desirably encode the mid-level semantics of different facial action parts.

1 Introduction

Facial expression analysis plays an important role in many computer vision applications, such as human-computer interaction and movie making. Many works have been done in the literature [1,2], but it remains unsolved. One of the key problems is how to represent different facial expressions. In the past decade, all kinds of local features have been exploited for facial expression analysis. However, making use of these hand-crafted local features might be essentially not good (if not wrong), considering that these features are also successfully exploited by face-based identity recognition methods. In principle, features for expression and identity recognition should be somehow exclusive.

To step out the trap, instead of manually designing local features, data-driven representation learning or deep learning is becoming popular more recently,

D. Cremers et al. (Eds.): ACCV 2014, Part IV, LNCS 9006, pp. 143–157, 2015.
DOI: 10.1007/978-3-319-16817-3_10

which emphasizes to hierarchically learn features that can be optimal to specific vision task. Among them, Convolutional Neural Network (CNN) [3] is one of the most successful ones for still image classification. Later on, it is further extended to 3D CNN [4] in order to deal with video-based action recognition problem. Our initial thought is applying CNN or 3D CNN directly to expression analysis, but we soon find it is even not better than hand-crafted features, e.g. HOG 3D [5] or LBP-TOP [6].

So, we realize that deep learning methods like CNN also need to be adapted to some new problems by incorporating the priors in the specific domain. In the case of expression analysis, studies in psychology have shown that expressions are invoked by a number of small muscles located around certain facial parts, e.g. eyes, nose, and mouth. These facial parts contain the most descriptive information for representing expressions. This observation brings us to the same spirit of the Deformable Part Model (DPM) [7], a state-of-the-art method in object detection. In DPM, an object is modeled by multiple parts in a deformable configuration and a bank of part filters can be simultaneously learned in a discriminative manner. The difference in our case is that the parts here are action parts, which dynamically change with the episode evolution of the expression.

With above ideas in mind, in this paper, we make an attempt to adapt 3D CNN for jointly localizing the action parts and learning part-based representations for expression analysis, by imposing the strong spatial structural constraints of the dynamic action parts. Fortunately, we found that the CNN framework has offered flexible structures to address the above problem: (1) CNNs have explicitly considered the spatial locality especially for 2D images or 3D videos, which can generate the underlying feature maps similar to HOG features used in DPM; (2) CNNs apply multiple trainable filters in each layer, which can be naturally incorporated with the deformation operations of part filters. Thus it is intuitive to embed such a deformation layer into the CNNs framework for learning these part locations as well as their representations.

To implement the above main ideas, i.e., achieve joint action part localization and part-based representation learning under CNN framework, we employ 3D CNN [4] as the basic model for its ability of motion encoding in multiple contiguous frames. Specifically, to adapt it for our goal, a bank of 3D facial part filters are designed and embedded in the middle layer of the networks (referring to Fig. 1), and the deformable action parts models are trained discriminatively under the supervision provided by class labels in the last layer. The deep networks increase the interactions among different learning components, thus we can obtain a globally optimized model.

2 Related Works

Many existing works on dynamic expression recognition attempt to encode the motion occurring in certain facial parts. One category of the methods is based on local spatio-temporal descriptors, e.g. LBP-TOP [6] and HOG3D [5]. The features extracted in local facial cuboid have possessed the property of repeatability, which makes it robust to the intra-class variation and face deformation.

However, such rigid cuboids can only capture low-level information that lacks of semantic meanings, and they can hardly represent the complex variations over those mid-level facial action parts. Another category of the methods attempt to encode the motion of facial action parts using a certain number of facial landmarks. For example, in [8,9], Active Appearance Model [10] and Constrained Local Model [11] are used to encode shape and texture variations respectively. However, it is difficult to achieve accurate landmarks (or action parts) detection under expression variations due to the large non-rigid deformation. In addition, all the methods mentioned above treat the feature learning separately without considering the final objective of classification, thus making the learned feature and model lacking of specificity and discriminative power.

Owing to the ability of organizing several functional components as cascaded layers into a unified network, the deep model is especially suitable for integrating the action parts detection, feature construction within the classifier learning procedure. For video-based classification tasks, 3D CNN [4] is shown to be one of the state-of-the-art models in action recognition field which considers the motion information encoded in multiple contiguous frames. However, except for the additional temporal convolutional operations, there is no structure designed specifically for locating or encoding semantic action parts, which makes it unsatisfactory for direct using in expression recognition task. Considering the structured property of human face, a DPM inspired deformable facial part model can also be learned for dynamic expression analysis. Meanwhile, [12] proposed to embed a deformation handling layer into the traditional 2D CNN for robust pedestrian detection. However, without consideration of temporal variations, this method cannot be directly applied to deal with video-based classification tasks.

To cope with the limitations in current works, in this paper we make two improvements: (1) We extend the traditional 2D deformable part model to 3D, which models dynamic motion in more complex videos rather than static appearance in simple still images. (2) We transform the binary detection model into multi-class classification model, and even continuous prediction model due to the regression capability of neural networks. Such adaptation enables us to accomplish expression intensity estimation and discrete expression recognition simultaneously.

3 Method

3.1 Overview

As mentioned above, our method is an adapted 3D CNN, which jointly takes into account two goals: localizing the facial action parts and learning part-based representations. Overall, our deep architecture is shown in Fig. 1. As can be seen, there are seven successive layers in our deep network:

Input video segments are n contiguous frames extracted from a certain expression video. The face in each frame is detected and normalized to the size of 64×64 pixels.

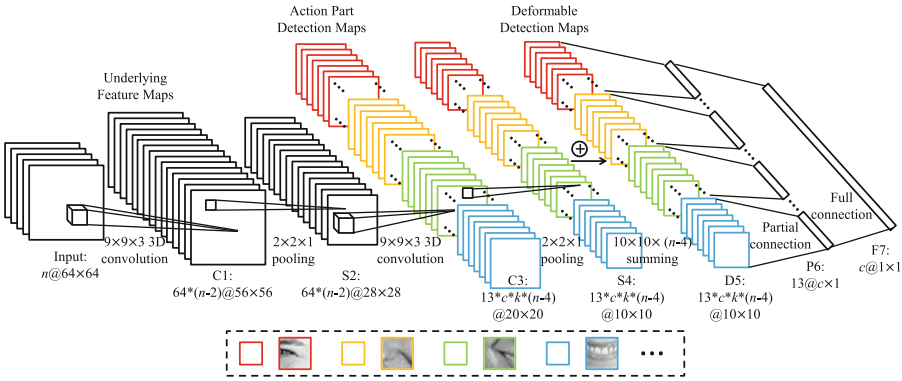


Fig. 1. An overview of the proposed deep architecture. The input n-frame video data is convolved with 64 **generic** 3D filters, and then mean-pooled to generate the underlying feature maps. Then the feature maps are convolved by $13 * c * k$ **specific** part filters to obtain the facial action part detection maps (where 13 is the number of manually defined facial parts, c is the number of classes, and k is the number of filters for one certain part in each class. The different colors represent the filter maps corresponding different parts). After the deformation maps weighting, the summed maps are processed by part-wise discriminative training to obtain the part-based estimation scores. Finally a full connection layer is used to predict the class label. **Best viewed in color** (Color figure online).

Underlying feature maps are obtained by convolving the input data using 64 spatio-temporal filters with the size of $9 \times 9 \times 3$. For translation invariance and dimension reduction, the filtered maps are then mean-pooled within non-overlapping $2 \times 2 \times 1$ spatial local region.

Action part detection maps are obtained by convolving the pooled underlying feature maps using a bank of class-specific 3D part filters. There are k filters for one certain part in each class to handle various manners of different people posing the same expression. Each detection map can be also regarded as response values of the whole face to a certain part filter. It is expected that the actual position of a detected action part arouses the largest response of the corresponding filter.

Deformable detection maps are obtained by summing up the part detection map and several deformation maps with learned weights. The deformation maps provide spatial constraints for detection maps according to the priors of facial configuration, which can fine-tune the detection scores and lead to a more reasonable result.

The partial connection layer concatenated to the deformable detection maps performs a part-wise discriminative learning for the part filters. As illustrated in Fig. 1, the different colors represent the detection maps corresponding to different facial action parts (We define 13 parts in this work). Totally 13 full connection structures are constructed for each part respectively for learning class-specific filters and outputs the part-based estimation scores. Finally we use a full

connection layer to predict the expression label. The whole model is optimized globally with back-propagation.

3.2 3D Convolution

In 2D CNNs, convolutions are applied on 2D images or feature maps to encode only spatial information. When processing video data, it is desirable to consider the motion variation in temporal dimension, i.e. multiple contiguous image frames. In [4], the 3D convolution is achieved by convolving the 3D kernels/filters on the cube constructed by image frames. Generally, we use the central symmetric g_{ijm} with respect to f_{ijm} to give an element-level formulation:

$$V_{ij}^{xyz} = \sigma(\sum_{m} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} V_{(i-1)m}^{(x+p)(y+q)(z+r)} \cdot g_{ijm}^{pqr} + b_{ij}), \tag{1}$$

where V_{ij}^{xyz} is the value at position (x, y, z) on the j-th feature map in the i-th layer. P_i, Q_i, R_i are the sizes of the 3D filters (R_i is for temporal dimension, and g_{ijm}^{pqr} is the (p, q, r)-th value of the filter connected to the m-th feature map in the $(i-1)$-th layer). The function $\sigma(x) = max(0, x)$ is the nonlinear operation used in our model, named Rectified Linear Units (ReLU) [13]. Such non-saturating nonlinearity can significantly reduce the training time compared with those saturating functions, e.g. *sigmoid* and *tanh* [14]. Extending the $*$ operation from 2D to 3D, we also have a simplified version:

$$V_{ij} = \sigma(\sum_{m} V_{(i-1)m} * f_{ijm} + b_{ij}) \tag{2}$$

3.3 Deformable Facial Action Parts Model

Taking the same spirit of the Deformable Part Model (DPM) [7], in our task, the model of a face with N parts (we set $N = 13$ in this work, see Fig. 2) can be defined as a set of part models $(P_1, P_2, \ldots, P_N)$, where $P_l = (F_l, v_l, s_l, d_l)$.

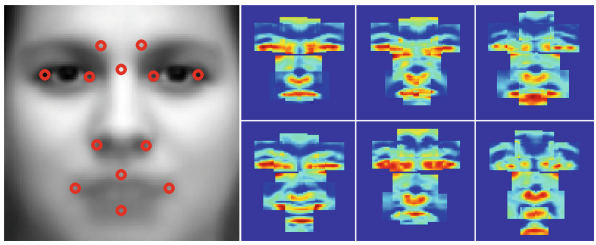


Fig. 2. The anchor positions of facial action parts (left) and illustration of the learned action parts filters for different expressions (right). For easy of visualization, we demonstrate the middle frame of the 3D filters. **Best viewed in color** (Color figure online).

Different from the original DPM, here $F_l = \{f_{[l,\theta]} | \theta = 1, 2, \ldots, c * k\}$ is a set of class-specific filters for detecting the l-th action parts of each expression respectively. v_l is a vector specifying the "anchor" positions for part l in the video, s_l is the size of the part detecting box, here the size is fixed by our 3D part filters in $C3$ layer, i.e. $9 \times 9 \times 3$. d_l is a weights vector of deformation maps specifying coefficients of a quadratic function defining deformation costs for possible placements of the part relative to the anchor.

Given the feature maps of a sample (i.e. the $S2$ layer), the l-th action part detection maps (i.e. the $C3$ layer) are obtained by convolving with a bank of part filters F_{3l} for response values. After the mean-pooling, the detection maps are summed with a set of weighted deformation maps to compute the deformable detection maps (i.e. the $D5$ layer). Note that here we process the operation for each 3D detection map separately corresponding to each single filter $f_{3[l,\theta]}$ in F_l. Formally, the scores on the map filtered by $f_{3[l,\theta]}$ in $D5$ layer is

$$\begin{aligned} D_{5[l,\theta]} &= S_{4[l,\theta]} - d_l \cdot \phi_d(dx_l, dy_l, dz_l) + b, \\ S_{4[l,\theta]} &= pool(C_{3[l,\theta]}), \\ C_{3[l,\theta]} &= \sigma(\sum_m (S_{2m} * f_{3[l,\theta]m}) + b_{3[l,\theta]})), \end{aligned} \tag{3}$$

where $[l, \theta]$ is the global index for the θ-th filter of the l-th part.

$$(dx_l, dy_l, dz_l) = (x_l, y_l, z_l) - v_l \tag{4}$$

gives the displacement of the l-th part relative to its anchor position, and

$$\phi_d(dx, dy, dz) = (dx, dy, dz, dx^2, dy^2, dz^2) \tag{5}$$

are deformation maps. Figure 3 shows an illustration of the deformable facial action part model. In general, the deformation cost is an arbitrary separable quadratic function of the displacements [7].

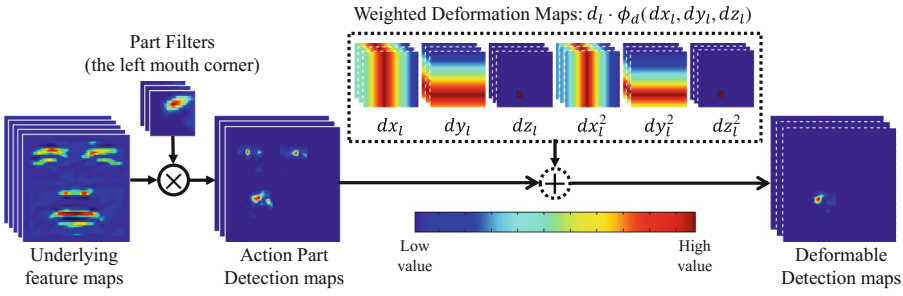


Fig. 3. An illustration of the deformable facial action part model. The part filters of left mouth corner may induce large response on the similar appearance position, e.g. eye corner. The spatial constraints provided by the deformation maps can effectively refine the detection maps. **Best viewed in color** (Color figure online).

In [12], only the maximum values of each deformable detection map are treated as the part scores for further prediction. However, in our multi-class recognition task, more diversified patterns are needed for describing each category particularly, rather than only tell "there is or not" in detection task. Therefore, the whole maps are retained for providing more information about the part filtered responses. Similar to [7,12], we conduct part-wise discriminative learning by partial connection to layer $P6$. Specifically, full connection structure are constructed for the maps of each part respectively and all the parameters in the part models (F_l, v_l, s_l, d_l) are optimized during the back-propagation.

4 Model Learning

4.1 Parameter Initialization

Training our deep model is a difficult task due to the millions of parameters. Therefore, we first initialize some important parameters and then update them all in the globally fine-tuning as in [15]. In this work, all the 3D convolution filters and the last two layers connection weights are chosen to be initialized.

Initialization of the filters. There are two kinds of filters in our model, i.e. the generic filters $\{f_{1m}\}$ and the specific part filters $\{f_{3[l,\theta]}\}$. Inspired by the work [16], we apply K-means clustering to learn centroids from the former feature maps and take them as the convolution filters. Specifically, we first learn 64 3D centroids from the input video, i.e. $\{f_{1m}\}$. Then we can obtain the $C1$ layer as

$$C_{1m} = \sigma(V_{input} * f_{1m} + b_{1m}). \tag{6}$$

The part filters $\{f_{3[l,\theta]}\}$ are learned from the pooled $S2$ layer. In the training set, we take the automatically detected $N = 13$ landmarks (as shown in Fig. 2, the initial anchor points are detected by SDM [17]) as the anchor positions of action parts, and sample the $9 \times 9 \times 3$ 3D cuboids around the anchors. The cuboids coming from the same position and same expression class are grouped up to learn k centroids, which are served as the class-specific part filters for layer $C3$. Totally there are $N * c * k$ filters in $\{f_{3[l,\theta]}\}$.

Initialization of the connection weights. After deformation handling layer $D5$, all the values of the same part, denoted as $D_{5[l,\cdot]}$, are fully connected to a subset of units in $P6$ layer, namely P_{6l}. We use W_{6l} to represent the connection weights corresponding to the l-th part, then

$$P_{6l} = \sigma(W_{6l}^T \ span(D_{5[l,\cdot]})). \tag{7}$$

where $span(\cdot)$ defines the operation of vectorization. Then the $P6$ is fully connected to $F7$:

$$F_7 = W_7^T P_{6[\cdot]}, \tag{8}$$

where $P_{6[\cdot]}$ represents the concatenated N part-based estimation scores, i.e. P_{6l}, in $P6$ layer. For initialization, we can directly use a linear transform for approximation. Therefore, the W_{6l} can be learned by a linear regression.

$$W_{6l} = Y^T span(D_{5[l,\cdot]})(span(D_{5[l,\cdot]})^T span(D_{5[l,\cdot]}) + \lambda I)^{-1} \tag{9}$$

where Y is the ground truth label matrix. Similarly, the W_7 can be learned as

$$W_7 = Y^T P_{6[\cdot]}(P_{6[\cdot]}^T P_{6[\cdot]} + \lambda I)^{-1} \tag{10}$$

4.2 Parameter Update

We update all the parameters after initialization by minimizing the loss function of square error

$$L(\mathcal{F}, \mathcal{D}, \mathcal{W}) = \frac{1}{2}||F_7 - Y||^2 = \frac{1}{2}e^2, \tag{11}$$

where $e = F_7 - Y$ is the error vector. $\mathcal{F} = \{F_1, \ldots, F_N\}$ and $\mathcal{D} = \{d_1, \ldots, d_N\}$ are part filters and weights vectors of deformation maps. $\mathcal{W} = \{\{W_{61}, \ldots, W_{6N}\}, W_7\}$ are connection matrices. The gradients of W_7 and P_{6l} can be computed by

$$\frac{\partial L}{\partial W_7} = P_{6[\cdot]} e^T, \tag{12}$$

$$\frac{\partial L}{\partial P_{6l}} = \frac{\partial L}{\partial P_{6[\cdot]}} \circ Mask_l = W_7\ e \circ Mask_l = \delta_{6l}, \tag{13}$$

where "$\circ$" represents element-wise multiplication and $Mask_1$ is a 0-1 vector to retain the connections only for the l-th part. For easy to express, we denote the gradient of P_{6l} as δ_{6l}. Then the gradients of W_{6l} and $D_{5[l,\cdot]}$ are

$$\frac{\partial L}{\partial W_{6l}} = span(D_{5[l,\cdot]})\delta_{6l}^T \circ I(P_{6l} > 0), \tag{14}$$

$$\frac{\partial L}{\partial D_{5[l,\cdot]}} = W_{6l}\delta_{6l} \circ I(P_{6l} > 0), \tag{15}$$

where $I(\cdot)$ is an index function to compute the derivative of ReLU. Given the gradient of $D_{5[l,\cdot]}$, we can obtain the the gradient of $D_{5[l,\theta]}$ at the same time by a simple reshape operation. The weights of deformation maps d_l can be updated according to its gradient

$$\frac{\partial L}{\partial d_l[t]} = \sum_{\theta} \frac{\partial L}{\partial D_{5[l,\theta]}} \circ \phi_d[t], \tag{16}$$

where $d_l[t]$ is the t-th component of the weights vector d_l and $\phi_d[t]$ is the t-th component of the deformation maps. Note that the $\partial L/\partial D_{5[l,\theta]}$ and $\phi_d[t]$ are both 3D feature maps. According to Eq. (3), we also have

$$\frac{\partial L}{\partial S_{4[l,\theta]}} = \frac{\partial L}{\partial D_{5[l,\theta]}} = \delta_{4[l,\theta]}, \tag{17}$$

then the gradient of $f_{3[l,\theta]}$ can be calculated as

$$\begin{aligned} \frac{\partial L}{\partial f_{3[l,\theta]}} &= \delta_{4[l,\theta]} \frac{\partial S_{4[l,\theta]}}{\partial C_{3[l,\theta]}} \frac{\partial C_{3[l,\theta]}}{\partial f_{3[l,\theta]}} \\ &= \sum_m S_{2m} * (up(\delta_{4[l,\theta]}) \circ I(C_{3[l,\theta]} > 0)). \end{aligned} \tag{18}$$

where $up(\cdot)$ is the up-sampling using the same definition in [18].

The gradient of the first convolutional layer f_{1m} can be calculated with the chain rule in the same way as $f_{3[l,\theta]}$. When obtain all the gradient, we can update the parameters using the stochastic gradient descent as in [15]. Take W_7 for example, the update rule of of W_7 in the k-th iteration is

$$\Delta^{k+1} = \alpha \cdot \Delta^k - \beta \cdot \epsilon \cdot W_7^k - \epsilon \cdot \frac{\partial L}{\partial W_7^k}, \tag{19}$$

$$W_7^{k+1} = \Delta^{k+1} + W_7^k, \tag{20}$$

where Δ is the momentum variable [19], ϵ is the learning rate and α, β are tunable parameters. In the training process, the learning rate is set as a fixed value 0.01.

5 Experiments

We evaluate our model on two posed expression datasets, CK+ [8], MMI [20], and a spontaneous dataset FERA [21] in four aspects: (1) visualization of the deformable part detection maps; (2) the loss of training/test sets before and after parameter updating; (3) qualitative results of expression intensity prediction; (4) quantitative results of average expression recognition rate and overall classification accuracy.

5.1 Data

CK+ database contains 593 videos of 123 different subjects, which is an extended version of CK database [22]. All of the image sequences vary in duration from 10 to 60 frames and start from the neutral face to the peak expression. Among these videos, 327 sequences from 118 subjects are annotated with the seven basic emotions (i.e. Anger (An), Contempt (Co), Disgust (Di), Fear (Fe), Happy (Ha), Sadness (Sa), and Surprise (Su)) according to FACS [23].

MMI database includes 30 subjects of both sexes and ages from 19 to 62. In the database, 213 image sequences have been labeled with 6 basic expressions, in which 205 are with frontal face. Different from CK+, the sequences in MMI cover the complete expression process from the onset apex, and to offset. In general, MMI is considered to be more challenging for the subjects usually wear some accessories (e.g. glasses, mustaches), and there are also large inter-personal variations when performing the same expression. The number of samples for each expression in CK+ and MMI are illustrated in Table 1.

FERA database is a fraction of the GEMEP corpus [24] that has been put together to meet the criteria for a challenge on facial AUs and emotion recognition. As the labels on test set are unreleased, we only use the training set for evaluation (Table 2). The training set includes 7 subjects, and 155 sequences have been labeled with 5 expression categories: Anger (An), Fear (Fe), Joy (Jo),

Table 1. The number of samples for each expression in CK+ and MMI database.

Expression	An	Co	Di	Fe	Ha	Sa	Su
CK+	45	18	59	25	69	28	83
MMI	31	–	32	28	42	32	40

Table 2. The number of samples for each expression in FERA database.

Expression	An	Fe	Jo	Sa	Re
FERA	32	31	30	31	31

Sadness (Sa), and Relief (Re). FERA is more challenging than CK+ and MMI because the expressions are spontaneous in natural environment.

We adopt the strictly person-independent protocols on both two databases for evaluation. In detail, experiments are performed based on 15-fold cross validation in CK+ and 20-fold cross validation in MMI, exactly the same as that in [25] for fair comparison. For FERA, as the labels on test set are unreleased, we adopt leave-one-subject-out cross-validation on the training set.

5.2 Evaluation of the Model

The deep model requires equal size of image cube, i.e. n frames as shown in Fig. 1. Given a T-frame video, we can pick up $T - n + 1$ cubes as the input data. For training samples, according to the expression varying manner in a sequence, we assign soft label values to the $T - n + 1$ training sample, i.e. video segments, using a gaussian function. For test samples, there is no need to know the ground truth of expression frames. After obtaining the $T - n + 1$ predict label vector, an aggregation strategy proposed in [26] is employed to generate the final recognition result of the whole video.

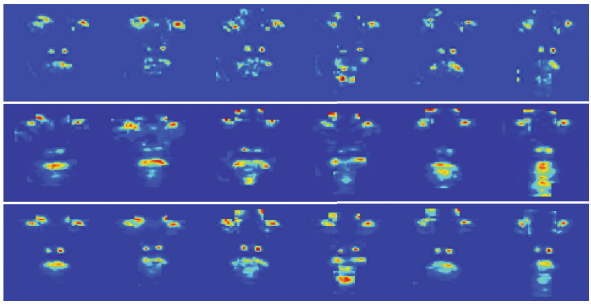
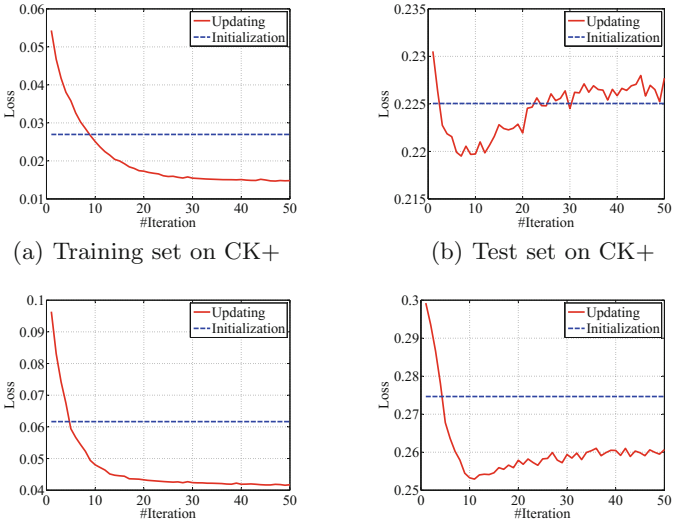


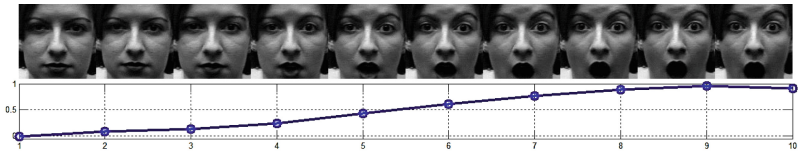
Fig. 4. Part detection maps of different part filters for different expressions. (We show the responses of all parts in one image by averaging the detection maps in the middle frame). **Best viewed in color** (Color figure online).



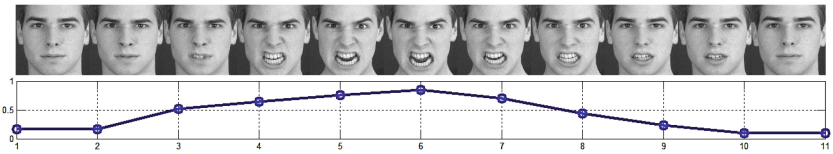
(a) Training set on CK+ (b) Test set on CK+

(c) Training set on MMI (d) Test set on MMI

Fig. 5. The loss of training and test sets on CK+ and MMI database.



(a) An example on CK+ database. (Surprise)



(b) An example on MMI database. (Angry)

Fig. 6. The expression intensity prediction results for two test sequences. The predicted scores are all for a video segments. For easy to visualization, we demonstrate the middle frame of each video segment to show the temporal variations of the expression.

After training the deep model, the part based representation can be obtained in $D5$ layer, i.e. the deformable detection maps. Each map is composed of the response values of a certain part filter, which depict various appearance patterns of different expressions. In Fig. 4, we provide a visualization of some selected deformable part detection maps learned by our model.

Moreover, to evaluate the learning ability of our deep model, we demonstrate the loss (defined in Eq. 11) of training/test set before and after parameter

updating in Fig. 5. As for validation purpose only, we conduct such experiments on one fold of CK+ and MMI respectively. In each figure, the blue curve is the loss of model using the initialized parameters for comparison. The red curve is the loss during the parameter updating, which shows a consistently decreasing trend on the training sets of both databases. However, it is easy to witness overfitting at a small number of iterations on the test sets, especially on CK+.

Our model can also provide the expression intensity prediction due to the regression ability of the neural networks. As presented in Sect. 5.1, a T-frame test sequence can generate $T - n + 1$ sub-segments for equal length inputs. Thus we can obtain $T - n + 1$ predict values for describing the changing of intensity during the whole expression process. We show some typical results of the intensity prediction along with the image/video data in Fig. 6.

5.3 Comparisons with Related Works

We compare our method, denoted by 3DCNN-DAP (Deformable Action Parts), with other related works under the same protocols adopted in [25]. Both average recognition rate and overall classification accuracy are measured. The results are listed in Tables 3 and 4 for CK+, Tables 5 and 6 for MMI, Tables 7 and 8 for FERA.

Specifically, to evaluate the most relevant work, the 3D CNN [4] fairly, we also conduct a similar parameter initialization using the same number of filters in each convolutional layer and a linear regression in the last full connection layer. The significant improvement shows that our deformable action part learning component has great advantages on task-specific feature representation.

Table 3. The average expression recognition rates on CK+ database.

Method	An	Co	Di	Fe	Ha	Sa	Su	Average
CLM [9]	70.1	52.4	92.5	72.1	94.2	45.9	93.6	74.4
AAM [8]	75.0	**84.4**	94.7	65.2	**100**	68.0	96.0	83.3
HMM [25]	–	–	–	–	–	–	–	83.5
ITBN [25]	**91.1**	78.6	94.0	83.3	89.8	76.0	91.3	86.3
HOG3D [5]	84.4	77.8	94.9	68.0	**100**	75.0	**98.8**	85.6
LBP-TOP [6]	82.2	77.8	91.5	72.0	98.6	57.1	97.6	82.4
3DCNN [4]	77.8	61.1	**96.6**	60.0	95.7	57.1	97.6	78.0
3DCNN-DAP	**91.1**	66.7	**96.6**	80.0	98.6	**85.7**	96.4	**87.9**

Table 4. The overall classification accuracy on CK+ database.

Method	CLM	AAM	ITBN	HOG3D	LBP-TOP	3DCNN	**3DCNN-DAP**
Accuracy	82.3	88.3	88.8	90.8	88.1	85.9	**92.4**

Table 5. The average expression recognition rates on MMI database.

Method	An	Di	Fe	Ha	Sa	Su	Average
HMM [25]	–	–	–	–	–	–	51.5
ITBN [25]	46.9	54.8	**57.1**	71.4	**65.6**	**62.5**	59.7
HOG3D [5]	61.3	53.1	39.3	78.6	43.8	55.0	55.2
LBP-TOP [6]	58.1	56.3	53.6	78.6	46.9	50.0	57.2
3DCNN [4]	58.1	21.9	25.0	83.3	53.1	**62.5**	50.7
3DCNN-DAP	**64.5**	**62.5**	50.0	**85.7**	53.1	57.5	**62.2**

Table 6. The overall classification accuracy on MMI database.

Method	ITBN	HOG3D	LBP-TOP	3DCNN	**3DCNN-DAP**
Accuracy	60.5	56.6	58.1	53.2	**63.4**

Table 7. The average expression recognition rates on FERA database.

Method	An	Fe	Jo	Sa	Re	Average
HOG3D [5]	43.8	33.3	74.2	54.8	48.4	50.9
LBP-TOP [6]	59.4	40.0	35.5	61.3	61.2	51.5
3DCNN [4]	34.4	26.7	64.5	51.6	54.8	46.4
3DCNN-DAP	50.0	58.1	73.3	51.6	48.4	**56.3**

Table 8. The overall classification accuracy on FERA database.

Method	HOG3D	LBP-TOP	3DCNN	**3DCNN-DAP**
Accuracy	51.0	51.6	46.5	**56.1**

6 Conclusions

In this paper, by borrowing the spirits of Deformable Part Models, we adapt 3D CNN to deeply learn the deformable facial action part model for dynamic expression analysis. Specifically, we incorporate a deformable parts learning component into the 3D CNN framework to detect special facial action parts under the structured spatial constraints, and obtain the deformable part detection maps to serve as the part-based representation for expression recognition. Such a deep model makes it possible to jointly localize the facial action parts and learn part-based representation. Impressive results beating the state-of-the-art are achieved on two challenging datasets.

To put it in another perspective, we have actually extended the deformable static part models to deformable dynamic part models under the CNN framework, which might also be validated by video-based event or behavior analysis.

In the future work, we will also try to consider more complex patterns of the action parts, e.g., of different sizes and shapes, or even with different time durations, to generate more flexible description of the facial expressions.

Acknowledgement. The work is partially supported by Natural Science Foundation of China under contracts nos. 61379083, 61272321, 61272319, and the FiDiPro program of Tekes.

References

1. Pantic, M., Rothkrantz, L.: Automatic analysis of facial expressions: the state of the art. IEEE T PAMI **22**, 1424–1445 (2000)
2. Zeng, Z., Pantic, M., Roisman, G., Huang, T.: A survey of affect recognition methods: audio, visual, and spontaneous expressions. IEEE T PAMI **31**, 39–58 (2009)
3. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**, 2278–2324 (1998)
4. Ji, S., Xu, W., Yang, M., Yu, K.: 3D convolutional neural networks for human action recognition. IEEE T PAMI **35**, 221–231 (2013)
5. Klaser, A., Marszalek, M.: A spatio-temporal descriptor based on 3D-gradients. In: BMVC (2008)
6. Zhao, G., Pietikainen, M.: Dynamic texture recognition using local binary patterns with an application to facial expressions. IEEE T PAMI **29**, 915–928 (2007)
7. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE T PAMI **32**, 1627–1645 (2010)
8. Lucey, P., Cohn, J., Kanade, T., Saragih, J., Ambadar, Z., Matthews, I.: The extended cohn-kanade dataset (ck+): a complete dataset for action unit and emotion-specified expression. In: CVPRW (2010)
9. Chew, S., Lucey, P., Lucey, S., Saragih, J., Cohn, J., et al.: Person-independent facial expression detection using constrained local models. In: FG (2011)
10. Cootes, T., Edwards, G., Taylor, C., et al.: Active appearance models. IEEE T PAMI **23**, 681–685 (2001)
11. Cristinacce, D., Cootes, T.F.: Feature detection and tracking with constrained local models. In: BMVC (2006)
12. Ouyang, W., Wang, X.: Joint deep learning for pedestrian detection. In: ICCV (2013)
13. Nair, V., Hinton, G.: Rectified linear units improve restricted boltzmann machines. In: ICML (2010)
14. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
15. Zhu, Z., Luo, P., Wang, X., Tang, X.: Deep learning identity preserving face space. In: ICCV (2013)
16. Coates, A., Ng, A., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: ICAIS (2011)
17. Xiong, X., De la Torre, F.: Supervised descent method and its applications to face alignment. In: CVPR (2013)
18. Bouvrie, J.: Notes on convolutional neural networks (2006)
19. Qian, N.: On the momentum term in gradient descent learning algorithms. Neural Netw. **12**, 145–151 (1999)

20. Valstar, M., Pantic, M.: Induced disgust, happiness and surprise: an addition to the MMI facial expression database. In: LRECW (2010)
21. Valstar, M.F., Mehu, M., Jiang, B., Pantic, M., Scherer, K.: Meta-analysis of the first facial expression recognition challenge. IEEE TSMCB **42**, 966–979 (2012)
22. Kanade, T., Cohn, J., Tian, Y.: Comprehensive database for facial expression analysis. In: FG (2000)
23. Ekman, P., Friesen, W.: Facial Action Coding System: A Technique for the Measurement of Facial Movement. Consulting Psychologists Press, Palo Alto (1978)
24. Bänziger, T., Scherer, K.R.: Introducing the geneva multimodal emotion portrayal (GEMEP) corpus. In: Scherer, K.R., Bänziger, T., Roesch, E.B. (eds.) Blueprint for Affective Computing: A Sourcebook, pp. 271–294. Oxford university Press, Oxford (2010)
25. Wang, Z., Wang, S., Ji, Q.: Capturing complex spatio-temporal relations among facial muscles for facial expression recognition. In: CVPR (2013)
26. Kanou, S., Pal, C., Bouthillier, X., Froumenty, P., Gülçehre, Ç., Memisevic, R., Vincent, P., Courville, A., Bengio, Y., Ferrari, R., et al.: Combining modality specific deep neural networks for emotion recognition in video. In: ICMI (2013)