Journal of Computers Vol. 31 No. 5, 2020, pp. 127-145
doi:10.3966/199115992020103105010

Deep Learning for Expressions Recognition on Small Dataset Using the Methods of Dual Transfer Learning and Feature Visualization

Junhuan Lin[1,2], Zhen Liu[1*], Chih-Chieh Hung[3], Ting Ting Liu[4], Yan Jie Chai[1]

[1] Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo, Zhejiang 315211

[2] Mechanical and Electrical Engineering College, Taizhou Vocational and Technical College, Taizhou, Zhejiang 318000

[3] College of Engineering, Tamkang University, Tamsui 25137

[4] College of Science and Technology, Ningbo University, Ningbo 315211
linjunh@qq.com

Received 16 April 2019; Revised 7 October 2019; Accepted 21 November 2019

Abstract. In this paper, we propose a dual transfer learning framework for image-based facial expressions recognition combining the deep convolutional neural networks(CNN) and feature visualization technique. The framework includes three steps. The first step is visualizing the features of BVLC's CNN to observe the pixels-level images reconstructed by the strongest activated neurons using deconvolutional method. As a result, some useful convolutional layers of the BVLC's CNN can be transferred to the next new targeting CNN immediately. Then the first transfer learning model of CNN is built up by concatenating the convolutional layers from BVLC's CNN to other convolutional layers. The second step is visualizing the features of the first transfer learning model after being trained on a medium dataset which is relevant to attributes of face. According to the results of feature visualization, the second transfer learning model can be built like the first steps. In the last step, the second transfer learning model is fine tuned on the CK+ dataset and used for recognizing the expressions. The testing results of classifying basic expressions demonstrate that our model based on dual transfer learning approach outperforms the current state-of-the-art works. Additionally, we also verify that our model is robust against interferences caused by various occlusions.

Keywords: convolutional neural network, feature visualization, transfer learning

1 Introduction

During the past decades, artificial intelligence has received more and more attentions from researchers and has been applied in many areas [1] to benefit life, where the intelligent system of human-computer interaction has played important roles [2]. The main goal of the intelligent system of human-computer interaction is to understand the users' intentions by acquiring the audio and visual information, in which it is indispensable to recognize the emotions of users. Humans often rely on facial expressions to show their emotions. Therefore, in order to build the intelligent system for emotionally interacting with users, researchers have devoted themselves to make the intelligent systems or agents automatically recognize the facial expressions. Some works [3-4] make use of the theories of FACS (Facial Action Coding System) proposed by Paul Ekman [5], in which all kinds of expressions can be comprised of several 'action units' (AUs), to extract the features of expressions. But these features based on AUs are dependent on the handcrafted points on the face. There are also methods to recognize expressions using information such as facial shape and texture features [6-8]. Lately, with the deep learning's prevailing in

[*] Corresponding Author

various computer vision, myriads of related works based on deep convolutional neural network(CNN) to classify the emotions of human are researched and outstanding results are achieved [9-11]. The CNN has been proved that the shallow layers can learn low-level visual features such as edges and corners while the deep layers can learn some semantic and abstract attributes such as object parts [12-13]. Khorrami P et al. [9] also demonstrate that the high-level layers in the CNN model trained on the dataset of expressional images can extract the features like facial action units, which is verified by using the method of feature visualization [12]. All above works inspire us to apply the CNN to the task of facial expressions recognition.

As is known, the CNN is highly dependent on the large datasets to avoid overfitting, whereas the current typical datasets for facial expressions recognition usually only contain less than 10 thousand images such as ck+ dataset. There is obvious limitation to directly implement the CNN trained on small datasets to classify the expressions. In this paper, we devote our attentions to solve two problems. One is to have the CNN applicable for recognizing facial expressions when just a small dataset of facial expressions is available and another is to select reusable layers from original CNN model to a new targeting CNN model when transfer learning is applied for settling the problem of training on small dataset. Therefore, we utilize the approach of dual transfer learning to guarantee the CNN performs well on the CK+ dataset [14]. In transfer learning process, we propose method of feature visualization derived from [12] to determine the transferable layers. To sum up, there are two main contributions or creativities in this paper:

(1) To apply the CNN on a small dataset, we propose dual transfer learning approach that transfers the features of CNN learned on ImageNet and CelebA dataset to the new targeting CNN model successively, which prevents the model trained on small dataset from overfitting and saves much training time.

(2) Using the feature visualization technique to check efficacy of convolutional layers of CNN on the next new targeting dataset guarantees that some appropriate convolutional layers are selected to be transferred.

The rest of the paper is organized as follows. Section 2 describes related works. Section 3 describes the proposed approach in detail. Section 4 describes the procedure of experiments and discusses the results. Finally, some conclusions are made in section 5.

2 Related Works

Most techniques of facial expressions recognition aim to classify the 7 basic emotions proposed by Ekman [5], which have been regarded as universe emotions across cultures and groups and are called neutral, happy, surprised, fear, angry, sad, and disgusted [15]. There are also some commercial developed systems used for classifying the basic emotions, such as Microsoft Cognitive Service Pack Facial [16], Affectiva [17]. All the above systems are evolving for general application and getting more and more powerful. Facial expressions of human reveal their emotions immediately when interacting with each other. Also eyebrows, lips, nose, mouth, muscles of the face are the principal features by which humans can understand emotions of the interlocutors. Artificial intelligent systems need to automatically recognize the facial expressions of users, the techniques of which are usually divided into two classes depending on whether the feature extraction are handcrafted or automatically learned on a deep neutral network. In the traditional techniques, there are three major steps including facial parts detection, features extraction, and expressions classification. Comparing to the traditional techniques, the deep learning is an end-to-end technique which processes information from the input of image to the output of classification directly via the deep neutral network learned on a big dataset.

For traditional techniques, there exists a variety of techniques to extract features on facial components. The technique of geometric feature focuses on the striking facial components and builds the feature vector for training based on geometric relationship between facial components. Deepak Ghimire et al. [18] propose a technique of feature extraction which depended on the geometric features from global facial region, the positon and angle of 52 facial landmark points, where the Euclidean distance and angle of each pair of landmarks per frame are computed and both the multi-class adaboost and SVM are applied for classifying facial expressions. The technique of statistical feature extracts features of appearance from global face region or different facial regions covering diverse information. S L Happy et al. [19] construct feature vector making use of the local binary pattern histogram of different block sizes from a global facial region, furthermore utilizing principal component analysis to reduce the redundant information and classifying various facial expressions. But the recognition accuracy is prone to reduce since it concerns all components as the same importance, ignoring the different levels of importance corresponding different facial regions. For example, eyes, and mouth provide more useful information than others to classifier. Deepak Ghimire et al. [20] divide the whole face region into different local regions and find out important local regions using an incremental search technique. Then the region-specific appearance features are extracted to reduce the dimensions of feature vector, which improves the recognition accuracy. The video-based technique extracts appearance features as spatial features and calculates the geometric displacements of facial landmarks between the current frame and previous frame as temporal features. Myunghoon Suk et al. [21] utilize the Active Shape Model (ASM) to fit landmarks on a face, and then extract the corresponding dynamic features which are generated by the displacements between frames of neutral and expression features, showing higher accuracy in the extended Cohn-Kanade (CK+) dataset than others.

Aside from the 2D-based techniques, 3D and dynamic 3D-based techniques are applied frequently in facial expressions recognition due to the shortage in 2D-based techniques resulted from intrinsic variations in pose and illumination. Hamit Soyel et al. [22] make use of six characteristic distances which are extracted from the distribution of 11 facial feature points from the given points in the BU-3DFE which is a 3D face dataset for facial expressions recognition, obtaining an average expression recognition rate of 91.3%. Hamit Soyel et al. [23] introduce a multitude of distances, representing open intensity of the eyes, the height of the eyebrows, and the position of the mouth, then Probabilistic Neural Network architecture is implemented to recognize the facial expressions. Facial expressions are recognized with an average recognition rate of 87.8%. But the 3D-based techniques outperform 2D-based techniques at the cost of high computational resources.

Usually, the traditional techniques adopt features and classifier by experts, and are highly subjected to humans' experiences. For many well-known handcrafted features, such as Hog [24], Gabor [25], dense Bow [26], Sift [27], etc. even though they are used in real-time systems because of lower computational complexity, they can't simultaneously optimize all properties of the system.

Over the last decade, with the emergence of the deep learning, there has been a large number of deep-learning algorithms applied to the field of computer vision, which can automatically undertake tasks of features extraction, classification, and recognition. CNN, one of deep-learning algorithms, comprises three types of layers: convolutional layer, pooling layer and fully connected layer. In the learning stage, CNN highly depends on big dataset, whose convolutional layers can extract both low-level and high-level features using unsupervised learning method [28]. For unsupervised learning, every single-layer model such as RBM or auto-encoder is pre-trained on unlabeled dataset, with which a deep model is stacked in layer-on-layer structure and is concatenated to a traditional classifier. Finally, the stacked model is fine tuned on small labeled dataset to classify objects [29]. Although the unsupervised learning offers an efficient training algorithm to the CNN on a big dataset, it is at the cost of time and inferior to supervised learning in labeled dataset. AlexNet [30] is firstly proposed as a supervised learning CNN model, which needs much fewer connection and parameters and is more easily trained without problems of vanishing gradient and overfitting on the large labeled datasets, and achieves spectacular results. Later on, many novel works [31-33] derived from AlexNet [30] continually improve the performance of the CNN. Breuer, R et al. [34] propose a CNN visualization technique to demonstrate that the CNN can be trained on dataset of facial expressions recognition. Heechul Jung et al. [35] combines two types of CNN to be an integration model to boost the performance of facial expressions recognition, where the one extracts temporal appearance features from image sequence and the other extracts temporal geometric features. Kaili Zhao et al. [36] unite region learning and multi-label learning to achieve facial action unit detection, in which a region layer uses feed-forward functions to induce important facial regions, and then promotes the learned weights to capture structural information of the face. Although CNN can extract both lower-level and semantic features, it tends to overfitting in training on a small dataset of facial expressions recognition. Additionally, training a completely new CNN model shall spend a large amount of time and resources. Transfer learning is a machine learning, where a model developed for a task is reused as the starting point for a model on a second task. In a CNN model, the low-level visual features and intermediate-level visual features learned on a large dataset for a task might be also useful for a new task. For a new CNN model to execute a new task, it is a shortcut to transfer some reusable layers from an existing CNN model trained on a large dataset. Jason Yosinski et al. [37] train an AlexNet on the ImageNet and find that the three low-level layers have common and reusable features across tasks.

Therefore, it is a feasible attempt to apply the transfer learning technique to facial expressions recognition, which can transfer some layers from an existing trained CNN model on a generic and large dataset to a new CNN model and furthermore fine tune the new model on a small dataset of facial expressions recognition. Yu Z et al. [38] pre-train the CNN on the facial expression recognition at first, then fine tune the model on the targeting dataset of SFEW2.0 by fixing the parameters of all convolutional layers and only updating the parameters at the fully connected layers and the model performs well. Levi G et al. [8] pre-train a CNN on CASIA Web face dataset whose images are transformed by technology of mapping the image intensities in 3D space then fine tune it on a limited emotion-labeled dataset to classify 7 basic expressions and obtain improved results. As is observed in papers [39-40], both targeting datasets are not large enough so that the CNNs tend to overfitting. Hong-Wei Ng et al. [40] propose a two-stage training process for a CNN, where the CNN is continuously trained on the dataset which is relevant to facial expressions at first based on the original CNN pre-trained on the ImageNet, then fine tune it on the targeting dataset. The testing results are better than the challenge baseline. M Peng et al. [41] present methods to recognize micro-expression in two provided datasets containing holdout-dataset recognition and composite dataset recognition. At first, the ResNet10 pre-trained on ImageNet dataset is trained on the large macro-expression dataset then fine tuned on the provided micro-expression dataset continuously. Experimental results confirm that the proposed method outperform baseline methods. As described above, no matter whether the CNN runs in training stage on big dataset or in fine tuning stage on small dataset, parameters of convolutional layers are either simultaneously updated or fixed. It might risk overfitting and cost much more time. If some layers in an existing trained CNN are supposed to be reusable for a new targeting CNN, then the parameters of those layers should needn't to be updated further. It would save many resources and much time to train CNN on computer and present more advantages of generalization.

If we can determine which layers of an existing CNN is reusable for a new targeting CNN, then those selected layers can be transferred to the new targeting CNN, whose parameters is fixed in both training and fine tuning stage. Matthew D. Zeiler et al. [12] provide us with an idea, feature visualization, which can visualize the activated neurons corresponding to the featured parts of the image on every convolutional layer by using deconvolutional method.

3 The Proposed Methods

3.1 The Framework of Our Method

In our methods, the BVLC's model [39] derived from AlexNet is adopted, which is trained on the ImageNet including more than one million images of various objects. Even though BVLC's model is based on ImageNet which isn't specially organized to classify emotions, the features learned by some low-level layers on ImageNet can still be shared with CK+ dataset, such as corners, edges. That means some knowledge learned by BVLC's model can be transferred to the task of facial expressions recognition on the CK+ dataset [42]. The overall pipeline of our proposed method is shown in Fig. 1 and is depicted as follows.

In the first step, to determine which layers from BVLC's model can be used immediately in new targeting CNN, we utilize the feature visualization technique to evaluate efficacy of individual layers of BVLC's model. Feature visualization technique shown in Fig. 2 can have individual activated neurons reconstructed in pixel-level space by using deconvolutional algorithm. In its forward process, inputting an original image from the CK+ dataset can activate neurons to generate a series of feature maps on individual layers. While in its backward process, neurons belonging to the top n activated neurons on feature maps are chosen to be reconstructed in pixel-level space showing what features the CNN has learned by using the layer-wise deconvolutional algorithm. For the individual layers, we can observe the discrepancies between the original image and the reconstructed images. Subsequently, we can check whether some convolutional layers in CNN can be transferred to the new targeting model.

In the second step, the first transfer learning CNN, some of whose layers are transferred from BVLC's model according to the judgment in the first step, is built then trained on the medium-scale CelebFaces Attributes (CelebA) dataset collected by Ziwei Liu et al. [43] including more than 200 thousand images. Some features learned from the CelebA dataset which is organized for recognizing the attributes on the

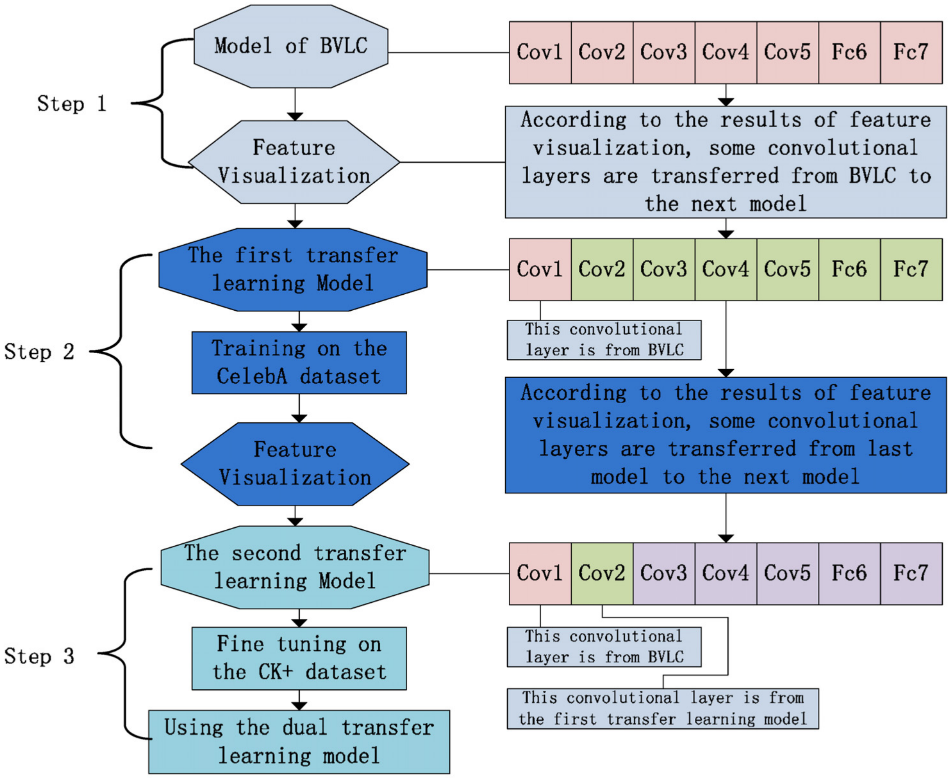
Step 1
Model of BVLC
Feature Visualization
Cov1 Cov2 Cov3 Cov4 Cov5 Fc6 Fc7
According to the results of feature visualization, some convolutional layers are transferred from BVLC to the next model
Step 2
The first transfer learning Model
Training on the CelebA dataset
Feature Visualization
Cov1 Cov2 Cov3 Cov4 Cov5 Fc6 Fc7
This convolutional layer is from BVLC
According to the results of feature visualization, some convolutional layers are transferred from last model to the next model
Step 3
The second transfer learning Model
Fine tuning on the CK+ dataset
Using the dual transfer learning model
Cov1 Cov2 Cov3 Cov4 Cov5 Fc6 Fc7
This convolutional layer is from BVLC
This convolutional layer is from the first transfer learning model

Fig. 1. The pipeline of our framework

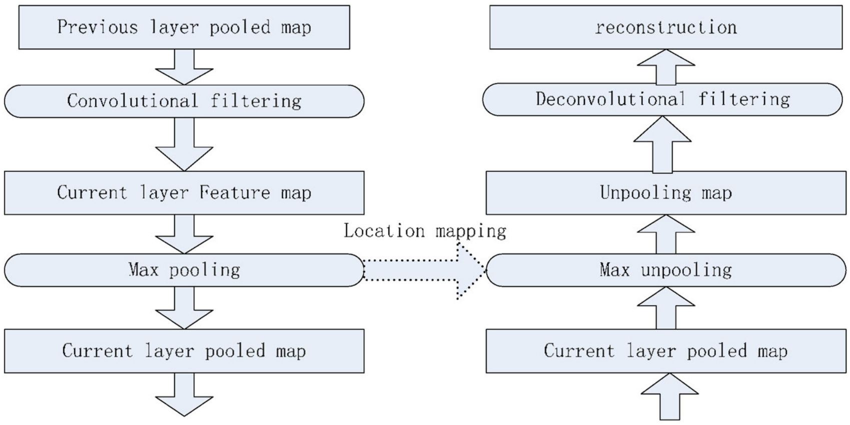

Fig. 2. The flowchart of feature visualization

face can be transferred to the task of classifying emotions due to the fact that there are common semantic features between attributes of face and expressions. Upon completion of training on CelebA dataset, we choose the convolutional layers which can be transferred from this CNN to another new targeting CNN in the same way as it operates in the first step.

In the last step, the second transfer learning CNN, some of whose layers are transferred from the first transfer learning CNN is fine tuned on the CK+ dataset to classify eight basic emotions such as happiness, anger, surprise, disgust, etc.

3.2 The Process of Feature Visualization

3.2.1 The Forward Process

As illustrated in Fig. 2, the feature visualization technique contains the forward process which can extract features by layer-wise convolution and pooling and the backward process which can reconstruct the specific neurons in pixel-level space by layer-wise deconvolution and unpooling. At first, the forward process is formulated as follows.

The structure of the CNN is presented in Fig. 3. For the p-th feature map of the l-th layer in CNN, the neurons are expressed in matrix as (1), where the size of feature maps are M×M, and correspondingly their kernels are expressed in matrix as (2) and the previous layer with k feature maps which are denoted with $Y_{pool}^{(l-1)(k)}$ as input for the l-th layer are expressed in matrix as (3), where the size is H×H.

$$Y_{conv}^{(l)(p)} = \begin{bmatrix} y_{conv(1)(1)}^{(l)(p)} & y_{conv(1)(2)}^{(l)(p)} & \cdots & y_{conv(1)(M)}^{(l)(p)} \\ y_{conv(2)(1)}^{(l)(p)} & y_{conv(2)(2)}^{(l)(p)} & \cdots & y_{conv(2)(M)}^{(l)(p)} \\ \vdots & \vdots & & \\ y_{conv(M)(1)}^{(l)(p)} & y_{conv(M)(2)}^{(l)(p)} & \cdots & y_{conv(M)(M)}^{(l)(p)} \end{bmatrix}. \tag{1}$$

$$W_k^{(l)(p)} = \begin{bmatrix} w_{11k}^{(l)(p)} & w_{12k}^{(l)(p)} & \cdots & w_{1Nk}^{(l)(p)} \\ w_{21k}^{(l)(p)} & w_{22k}^{(l)(p)} & \cdots & w_{2Nk}^{(l)(p)} \\ \vdots & & & \\ w_{N1k}^{(l)(p)} & w_{N2k}^{(l)(p)} & \cdots & w_{NNk}^{(l)(p)} \end{bmatrix}. \tag{2}$$

$$Y_{pool}^{(l-1)(k)} = \begin{bmatrix} y_{pool(1)(1)}^{(l-1)(k)} & y_{pool(1)(2)}^{(l-1)(k)} & \cdots & y_{pool(1)(H)}^{(l-1)(k)} \\ y_{pool(2)(1)}^{(l-1)(k)} & y_{pool(2)(2)}^{(l-1)(k)} & \cdots & y_{pool(2)(H)}^{(l-1)(k)} \\ \vdots & & & \\ y_{pool(H)(1)}^{(l-1)(k)} & y_{pool(H)(2)}^{(l-1)(k)} & \cdots & y_{pool(H)(H)}^{(l-1)(k)} \end{bmatrix}. \tag{3}$$

$Y_{conv}^{(l)(p)}$ deriving from convolving arithmetic between $W_k^{(l)(p)}$ and $Y_{pool}^{(l-1)(k)}$ is detailed as formula (4), where size of kernels are N×N, the stride is S and P_{conv} is the number of padding for convolution. Then the pooling layer $Y_{pool}^{(l)(p)}$ with size of U×U as formula (5) is calculated from $Y_{conv}^{(l)(p)}$ by finding the maximum neuron of individual pooling region depicted as formula (6), where size of pooling is R×R, the stride is T and P is the number of padding for pooling. To avoid losing location information in pooling, it is necessary to mark the indexes of the maximal neuron in the individual pooling region, which is denoted with $r_{\max(i)}^{(l)(p)}, r_{\max(j)}^{(l)(p)}$ expressed in (7).

$$\begin{aligned} y_{conv(j)(i)}^{(l)(p)} &= f\left(\sum_{k=1}^{M}\left(sum\left(W_k^{(l)(p)} \cdot (Y_{pool}^{(l-1)(k)})^T\right)\right) + b^{(l)(p)}\right) \\ &= f\left(\sum_{k=1}^{M} sum\left(\begin{bmatrix} w_{11k}^{(l)(p)} & w_{12k}^{(l)(p)} & \cdots & w_{1Nk}^{(l)(p)} \\ w_{21k}^{(l)(p)} & w_{22k}^{(l)(p)} & \cdots & w_{2Nk}^{(l)(p)} \\ \vdots & & & \\ w_{N1k}^{(l)(p)} & w_{N2k}^{(l)(p)} & \cdots & w_{NNk}^{(l)(p)} \end{bmatrix} \cdot \begin{bmatrix} y_{pool(1+(j-1)\cdot S)(1+(i-1)\cdot S)}^{(l-1)(k)} & \cdots & y_{pool(1+(j-1)\cdot S)(N+(i-1)\cdot S)}^{(l-1)(k)} \\ y_{pool(2+(j-1)\cdot S)(1+(i-1)\cdot S)}^{(l-1)(k)} & \cdots & y_{pool(2+(j-1)\cdot S)(N+(i-1)\cdot S)}^{(l-1)(k)} \\ \vdots & & \\ y_{pool(N+(j-1)\cdot S)(1+(i-1)\cdot S)}^{(l-1)(k)} & \cdots & y_{pool(N+(j-1)\cdot S)(N+(i-1)\cdot S)}^{(l-1)(k)} \end{bmatrix}^T\right) + b^{(l)(p)}\right) \end{aligned} \tag{4}$$

Where $j,i = 1,2,3 \cdots M$, $M = (H - N + P_{conv})/S + 1$.

$$Y_{pool}^{(l)(p)} = \begin{bmatrix} y_{pool(1)(1)}^{(l)(p)} & y_{pool(1)(1)}^{(l)(p)} & \cdots & y_{pool(1)(1)}^{(l)(p)} \\ y_{pool(2)(1)}^{(l)(p)} & y_{pool(2)(2)}^{(l)(p)} & \cdots & y_{pool(2)(U)}^{(l)(p)} \\ \vdots & & & \\ y_{pool(U)(1)}^{(l)(p)} & y_{pool(U)(2)}^{(l)(p)} & \cdots & y_{pool(U)(U)}^{(l)(p)} \end{bmatrix}. \tag{5}$$

$$y_{pool(j)(i)}^{(l)(p)} = MAX\left(\begin{bmatrix} y_{conv(1+(j-1)\cdot T)(1+(i-1)\cdot T)}^{(l)(p)} & y_{conv(1+(j-1)\cdot T)(2+(i-1)\cdot T)}^{(l)(p)} & \cdots & y_{conv(1+(j-1)\cdot T)(R+(i-1)\cdot T)}^{(l)(p)} \\ y_{conv(2+(j-1)\cdot T)(1+(i-1)\cdot T)}^{(l)(p)} & y_{conv(2+(j-1)\cdot T)(2+(i-1)\cdot T)}^{(l)(p)} & \cdots & y_{conv(2+(j-1)\cdot T)(R+(i-1)\cdot T)}^{(l)(p)} \\ \vdots & \vdots & & \\ y_{conv(R+(j-1)\cdot T)(1+(i-1)\cdot T)}^{(l)(p)} & y_{conv(R+(j-1)\cdot T)(2+(i-1)\cdot T)}^{(l)(p)} & \cdots & y_{conv(R+(j-1)\cdot T)(R+(i-1)\cdot T)}^{(l)(p)} \end{bmatrix}\right). \qquad \textbf{(6)}$$

Where $j,i = 1,2,3\cdots U$， $U = (M - N_{pool} + P_{pool})/T + 1$.

$$r_{\max(i)}^{(l)(p)}, r_{\max(j)}^{(l)(p)} = \arg MAX\left(\begin{bmatrix} y_{conv(1+(j-1)\cdot T)(1+(i-1)\cdot T)}^{(l)(p)} & \cdots & y_{conv(1+(j-1)\cdot T)((i-1)\cdot T+R)}^{(l)(p)} \\ \vdots & \cdots & \vdots \\ y_{conv((j-1)\cdot T+R)(1+(i-1)\cdot T)}^{(l)(p)} & \cdots & y_{conv((j-1)\cdot T+R)((i-1)\cdot T+R)}^{(l)(p)} \end{bmatrix}\right). \qquad \textbf{(7)}$$

Where $i,j = 1,2,3\cdots U$， $U = (M - N_{pool} + P_{pool})\big/T + 1$.

3.2.2 The Backward Process

In backward process, an arbitrary neuron $y_{pool(j)(i)}^{(l)(p)}$ in the l-th layer on the p-th feature map can be reconstructed into original pixel-level space through iterating arithmetic with unpooling and deconvolution as formula (8) and formula (9)-(13) respectively, where f^{-1} is activation reverse function, b are the bias values, $y_{extension(j)(i)}^{(l)(p)}$ are extended from $y_{conv(n)(m)}^{(l)(p)}$ with the regulation as formula (10) and $y_{pad(b)(a)}^{(l)(p)}$ are formed by padding with formula (11). Subsequently, the neurons $y_{pool(j)(i)}^{(l-1)(k)}$ in the (l-1)-th layer on the k-th feature map are computed from convolution between $Y_{pad}^{(l)(p)}$ comprising of $y_{pad(b)(a)}^{(l)(p)}$ and $W_k^{(l)(p)}$ comprising of $w_{(j)(i)k}^{(l)(p)}$.

$$y_{conv(n)(m)}^{(l)(p)} = \begin{cases} y_{pool(j)(i)}^{(l)(p)} & n = r_{\max(j)}^{(l)(p)}, m = r_{\max(i)}^{(l)(p)} \\ 0 & n \neq r_{\max(j)}^{(l)(p)}, m \neq r_{\max(i)}^{(l)(p)} \end{cases} \qquad (8)$$

Where $n \in \left[1+(j-1)\cdot T, (j-1)\cdot T + R\right]$, $m \in \left[1+(i-1)\cdot T, (i-1)\cdot T + R\right]$, $m, n = 1, 2, 3\cdots M$, $i, j = 1, 2, 3\cdots U$.

$$y_{conv(n)(m)}^{(l)(p)} = f^{-1}\left(y_{conv(n)(m)}^{(l)(p)}\right) - b^{(l)(p)} \qquad \textbf{(9)}$$

$$y_{extension(j)(i)}^{(l)(p)} = \begin{cases} y_{pool(n)(m)}^{(l)(p)}, when \quad j = N\cdot(n-1)+1, i = N\cdot(m-1)+1 \\ 0, others \end{cases}， j = 1,2\cdots m\cdot N， i = 1,2\cdots n\cdot N \qquad \textbf{(10)}$$

$$y_{pad(b)(a)}^{(l)(p)} = \begin{cases} y_{extension(j)(i)}^{(l)(p)}, b = j + P_{pad}, a = i + P_{pad} \\ 0 \quad \begin{matrix} b = 1,2\cdots P_{pad}, N\cdot n+1, N\cdot n+2\cdots, N\cdot n + P_{pad} \\ a = 1,2\cdots P_{pad}, N\cdot m+1, N\cdot m+2\cdots, N\cdot m + P_{pad} \end{matrix} \end{cases}， j = 1,2\cdots m\cdot N， i = 1,2\cdots n\cdot N \qquad \textbf{(11)}$$

$$Y_{pool}^{(l)(p)} = \sum_{p=1}^{P} conv2\left(Y_{pad}^{(l)(p)}, W_k^{(l)(p)}\right) \qquad \textbf{(12)}$$

$$y_{pool(j)(i)}^{(l-1)(k)} = \sum_{p=1}^{P}\left(\begin{bmatrix} y_{pad(j)(i)}^{(l)(p)} & y_{pad(j)(i+1)}^{(l)(p)} & \cdots & y_{pad(j)(i-1+N)}^{(l)(p)} \\ y_{pad(j+1)(i)}^{(l)(p)} & y_{pad(j+1)(i+1)}^{(l)(p)} & \cdots & y_{pad(j+1)(i-1+N)}^{(l)(p)} \\ \vdots & & & \vdots \\ y_{pad(j-1+N)(i)}^{(l)(p)} & y_{pad(j-1+N)(i+1)}^{(l)(p)} & \cdots & y_{pad(j-1+N)(i-1+N)}^{(l)(p)} \end{bmatrix} \cdot \begin{bmatrix} w_{(N)(N)(k)}^{(l)(p)} & w_{(N)(N-1)(k)}^{(l)(p)} & \cdots & w_{(N)(1)(k)}^{(l)(p)} \\ w_{(N-1)(N)(k)}^{(l)(p)} & w_{(N-1)(N-1)(k)}^{(l)(p)} & \cdots & w_{(N-1)(1)(k)}^{(l)(p)} \\ \vdots & & & \\ w_{(1)(N)(k)}^{(l)(p)} & w_{(1)(N-1)(k)}^{(l)(p)} & \cdots & w_{(1)(1)(k)}^{(l)(p)} \end{bmatrix}\right) \textbf{(13)}$$

4 Experiments

4.1 The First Transfer Learning Process

4.1.1 Selecting a Deep Convolutional Neural Network

We choose the BVLC's model as a parental structure of network throughout this paper. BVLC's model contains eight learned layers, five of which are convolutional layers and three of which are fully-connected layers. The structure is summarized in Fig. 3.

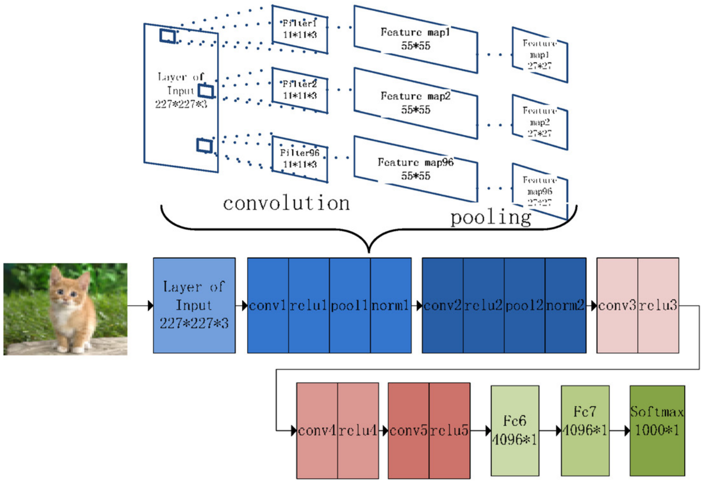

Fig. 3. An illustration of the structure

The structure of original networks has been rectified in BVLC's model where the first convolutional layer's input is the 227*227*3 inputting image and it outputs the 96 feature maps of size 55*55 with the use of 96 kernels of size 11*11*3 and a stride of 4 pixels then is pooled by using kernels of size 3*3 and a stride of 2 and subsequently yields 96 feature maps of size 27*27. The second convolutional layer is convolved with the 96 feature maps of size 27*27 using 256 kernels of size 5*5*48. The third, fourth, and fifth convolutional layers are connected without pooling and normalizing layers. The total internal structure is depicted in detail in Table 1.

Table 1. Structure of the deep convolutional neural networks

		1	2	3	4	5	6	7	8
layer	input	conv	pool	conv	pool	conv	conv	conv	pool
kernel	227*227*3	11*11*3*96	3*3*96	5*5*256	3*3*256	3*3*384	3*3*384	3*3*256	3*3*256
stride		4	2	0	2	0	0	0	2
pad		0	0	2	0	1	1	1	0

4.1.2 Feature Visualization

In a trained deep convolutional neural network, a given image pattern could be mapped to feature maps on every convolutional layer hierarchically in which the corresponding filters extract the higher-level features from output of their pre-layer. To understand what features have been learned on every convolutional layer for a given pattern, instead of mapping the pixels to feature maps of convolutional layers, we make use of the reverse course based on deconvolutional algorithm which is programmed in Matlab to remap top n activated neurons in the individual feature maps of convolutional layers to the pixel-level space.

In this paper, deep convolutional neural network is the BVLC's model trained on ImageNet. When using the method of feature visualization proposed in section 3.2, we can observe the results of visualizing every convolutional layer presented in Fig. 4, Fig. 5, Fig. 6, Fig. 7, Fig. 8.

These images in Fig. 4 are from the results of visualizing features on the first convolutional layer. The image in (a) is an original image from CK+ dataset. And the image in (b) shows contour that is equivalent to the effect of edge detection, which is reconstructed by all of the neurons in 96 feature maps using the method of feature visualization. The image in (c) shows the oblique slim polylines, which is reconstructed by all of the neurons in the first feature map. That means the first kernel filter has learned a low-level feature, the oblique slim polyline. Likewise, the image in (d) shows the vertical thick lines, which are reconstructed by all of the neurons in the fourth feature map in our model. The images in (e) and in (f) show the similar effects. The images in (g) and in (h) show some objects of eyes, nose and mouth all belonging to significant features of face, which are reconstructed by the top 1 and top 10 activated neurons of all the feature maps respectively. According to the results, we can conclude that the strongest activated neurons are activated by the most significant features of face correspondingly. The results corroborate the deep learning theories that the low-level convolutional layers can learn low-level features such as edges and corners meanwhile the strongest activated neurons are responded to the most significant features. Even though the first convolutional layer in BVLC's model has never been trained on CK+ dataset, but its abstracted low-level features can be shared across datasets.

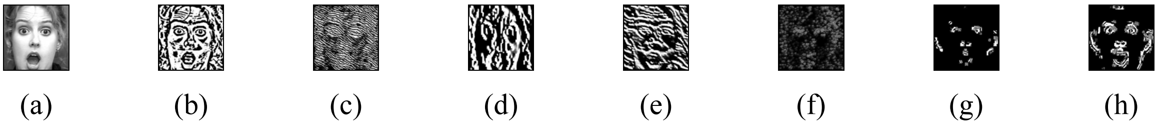

Fig. 4. The results of visualizing the first convolutional layers in BVLC's model

These images in Fig. 5 are from the results of visualizing features on the second convolutional layer. The image in (b) shows the head portrait which is reconstructed by all of the neurons in 256 feature maps using the method of feature visualization. As is presented, it has no edges as clear as in Fig. 4. The images in Fig. 5(c), Fig. 5(d), Fig. 5(e), Fig. 5(f), Fig. 5(g) show insignificant parts which are reconstructed by all the neurons of the fourth, thirteenth, thirty fifth and ninety fourth feature maps respectively. The image in (h) shows a blurred face which is reconstructed by the top 1 activated neurons of all the feature maps. As is seen, in contrast to the corresponding image in Fig. 4, the second convolutional layer contains so many insignificant parts of face that it can't be transferred to the new targeting CNN.

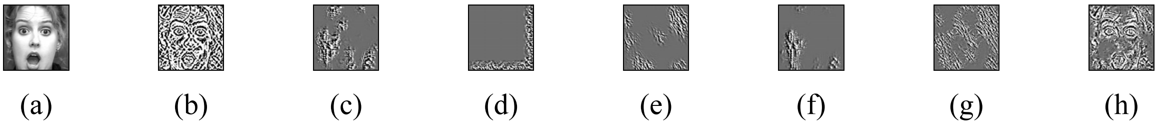

Fig. 5. The results of visualizing the second convolutional layers in BVLC's model

These images in Fig. 6 are from the results of visualizing features on the third convolutional layer following the same process.

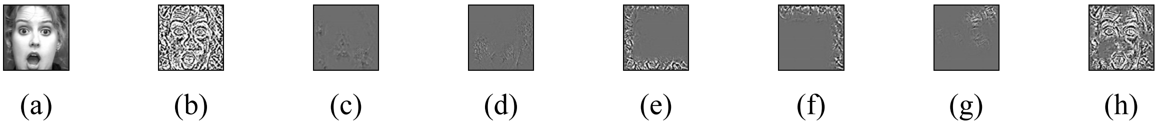

Fig. 6. The results of visualizing the third convolutional layers in BVLC's model

These images in Fig. 7 are from the results of visualizing features on the fourth convolutional layer following the same process.

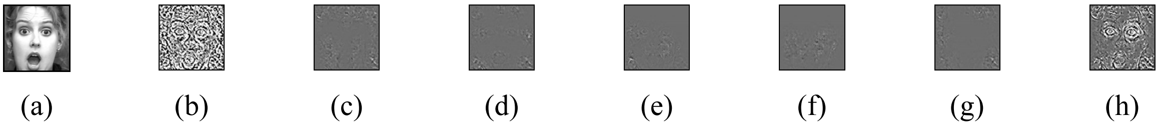

Fig. 7. The results of visualizing the four convolutional layers in BVLC's model

These images in Fig. 8 are from the results of visualizing features on the fifth convolutional layer following the same process.

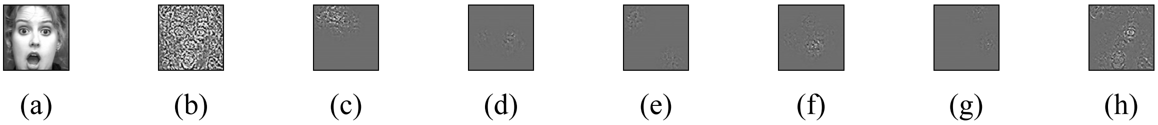

Fig. 8. The results of visualizing the fifth convolutional layers in BVLC's model

Observing the above results, with the convolutional layers going deeper, the images in Fig. 8(b) become more disordered meanwhile the images in Fig. 8(h) become more blurred. Hence, for the BVLC's model, only the first convolutional layer can be transferred to the next targeting CNN model, which is called the first transfer learning model.

4.2 The Second Transfer Learning Process

4.2.1 Training on the CelebA Dataset

Data preparation. The CelebA dataset is collected by Multimedia Lab affiliated to the Chinese University of Hong Kong, which contains 202,599 images from website and all the images are aligned and cropped to encase the face region in the same size 178*218. Additionally, every sample is annotated with 40 attribution labels about facial features such as arched-Eyebrows, bald, big-Lips, wearing-Earrings, etc. In order to match BVLC's model, it is necessary to preprocess the data. First, the images are cropped again and resized to 227*227 to match the input size. Second, the 40 attributions are annotated by 1 and 0 recoded from 1 and -1 of original code which denote whether a given image presents corresponding attributions in face respectively. Third, to reduce the training range and time, we only adopt 12 of 40 attributions which are closely relevant to expressions. The 12 attributions are summarized in Table 2 with an image of sample.

Table 2. 12 Attributions of face selected from CelebA dataset

Original image	attributions			
	Arched-Eyebrows	Bags-Under-Eyes	Big-Lips	Big-Nose
	1	0	0	0
	attributions			
	Oval-Face	High-Cheekbones	Mouth-Slightly-Open	Narrow-Eyes
code	1	0	0	0
	attributions			
	Pointy-Nose	Ros-Cheeks	Smiling	Bushy-Eyebrows
	0	0	1	0

Training. The original BVLC's model has been designed to classify 1000 classes of objects where every sample only has single label. However, when training on the CelebA dataset, we must face a challenge that every sample possesses 12 labels. Therefore, the internal structure of the first transfer learning model, which is inherited from the parental BVLC's model, needs modifying to deal with the task of multi-labels. Certainly, we hold the convolutional layers unchangeable then replicate the full-connect layers by 12 copies which are all connected to the last convolutional layers in BVLC's model in common. The number of neurons in the first and second full-connect layers are reset to 128 and 2 revised from original 1024 and 1000 respectively. The architecture is summarized in Fig. 9.

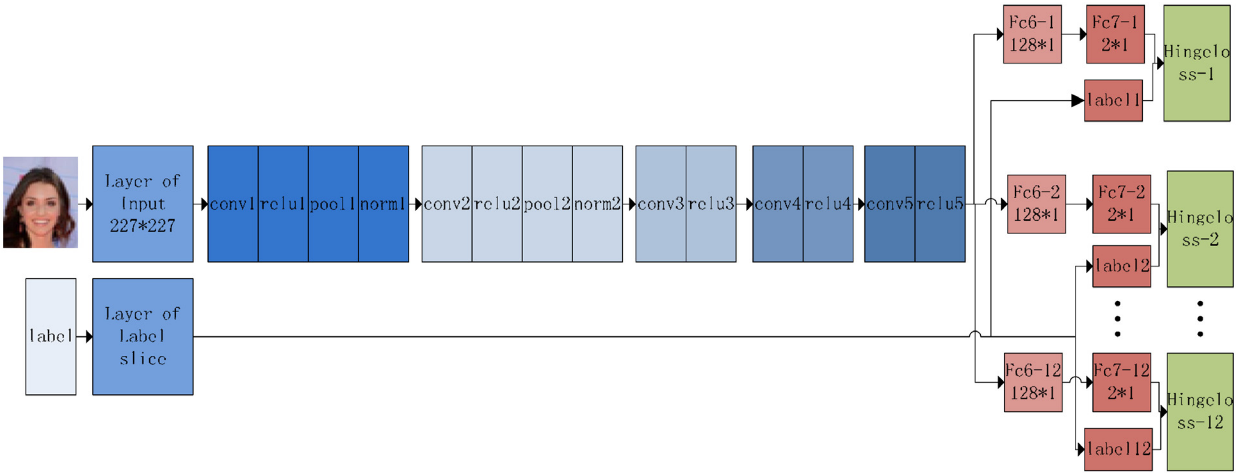

Fig. 9. The structure of the model for being trained on the CelebA

As is shown in the Fig. 9, the task of multi-labels is simplified into 12 two-classes classifying tasks with single label, which performs on the platform of caffe developed by Jia, Yangqing et al. [39]. The classification layer in this model adopts SVM taking the hinge loss as the loss function for training. The training program is executed on the workstation with 10G GPU, 16G RAM. Some hyper-parameters of model are summarized in Table 3. In the training phase, the first convolutional layer is free-trained and the parameters of other layers are filled with BVLC's model.

Table 3. Hyperparameters of model dataset

Parameters for solver	base_lr	Min_batch	lr_policy	gamma	stepsize
Value	0.0001	100	step	0.1	1000
Parameters for solver	max_iter	momentum	weight_decay	Optimized algorithm	
Value	100000	0.9	0.0005	SGD	

4.2.2 Feature Visualization

Using the trained model above, now we make use of the feature visualization to observe the results that the activated neurons on feature maps of every convolutional layer has been mapped in pixel-level space inversely. The results are shown in Fig. 10, Fig. 11, Fig. 12 and Fig. 13.

These images in Fig. 10 are from the results of visualizing features on the second convolutional layer. The image in (b) is reconstructed by all of the neurons in 256 feature maps. Compared to the image in Fig. 10(c) which is the counterpart in the first transfer learning process, the edges of the contour get more distinct. The image in Fig. 10(d), (e) and Fig. 10(f) are reconstructed by all of the neurons in the 19th, 56th and 151th feature maps respectively in this model, showing clearly that these kernels of feature map has learned the distinctive features of face very well. The image in Fig. 10(g) is reconstructed by the top 1 activated neuron of all the feature maps. Compared to the image in Fig. 10(h) which is the counterpart in the first transfer learning process, it indicates that the second convolutional layer inclines to be only sensitive to eyes and mouth which are related to expressions.

(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 10. The results of visualizing the second convolutional layers in the first transfer learning model

These images in Fig. 11 are from the results of visualizing features on the third convolutional layer. The image in (b) is reconstructed by all of the neurons in 256 feature maps and the image in Fig. 11(c) is the counterpart in the first transfer learning proess. Both two images have no clear edges of contour. The image in Fig. 11(d) is reconstructed by the top 1 activated neuron of the 100th feature maps failing to learn some significant features. The images in Fig. 11(e) and in Fig. 11(f) are reconstructed by top 10 and top 1 neurons of the 193th feature map respectively. When judging the distinctions between the two images, we can find that the most significant features such as eyes, mouth are not mapped by top 1 activated neuron but by other top 9 activated neurons, which deviates over the goal of deep learning. The image in Fig. 11(g) is reconstructed by the top 1 activated neuron of all the feature maps and the image in Fig. 11(h) is the counterpart in the first transfer learning. Both two images show there are some insignificant features to respond to the top 1 activated neuron of feature maps.

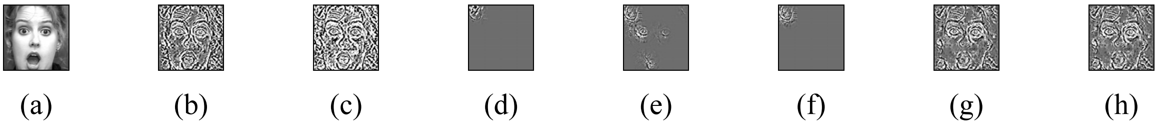

Fig. 11. The results of visualizing the third convolutional layers in the first transfer learning model

These images in Fig. 12 are from the results of visualizing features on the fourth convolutional layer. The images are arranged in the same order as Fig. 10 and Fig. 11.

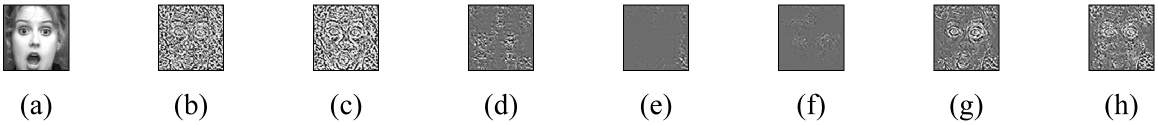

Fig. 12 The results of visualizing the fourth convolutional layers in the first transfer learning model

These images in Fig. 13 are from the results of visualizing features on the fifth convolutional layer. The images are arranged in the same order as Fig. 10 and Fig. 11.

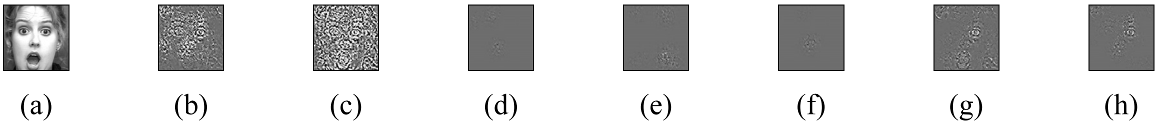

Fig. 13. The results of visualizing the fifth convolutional layers in the first transfer learning model

Observing the results from Fig. 10, Fig. 11, Fig. 12 and Fig. 13, we can reach the same conclusion that the deeper the convolutional layer is in the model, the more blurred the learned features become. Summing up from above results, only the second convolutional layer which has learned second-level features from CelebA dataset can be transferred to the second transfer learning model.

4.3 Fine Tuning on the CK+ Dataset

4.3.1 Data Preparation

The CK+ dataset is collected by Lucey et al. [14] containing 593 sequences across 123 subjects, 327 of which are labeled by 8 emotions including neutral, anger, contempt, disgust, fear, happy, sadness and surprise which are encoded from 0 to 7 respectively. All the sequences are sampled from the neutral face to the peak expressions and one labeled sequence corresponds to one emotion. However, the neutral face images included in one certain sequence are also regarded as the same emotional samples of this sequence, which would definitely harm the training process. So we single out neutral face images from the sequences and get them relabeled as neutral emotion.

The image should be preprocessed before being trained on the dataset. At first, redundant background outside of face box is cropped off using a common face detecting algorithm. Then the images are resized to 227*227 to match the input size of deep learning model. Finally, the gray images should be transformed into RGB data. We simply transform the gray data into RGB by replicating three copies from gray data.

Excluding the unlabeled sequences, the total number of images amounts to 1875, which are divided randomly into three groups including 1400 image for training data, 200 images for testing data and the others for validation data.

4.3.2 Fine Tuning

The second transfer learning model should be modified to classify 8 classes of emotions. Except for the first convolutional layer and second convolutional layer which are transferred from the two previous models, the other convolutional layers continue to be fine tuned on the CK+ dataset. In the training phase, all the convolutional layers are filled with parameters from previous model, meanwhile the first and second convolutional layers are fixed without being updated. The process is realized on the platform of caffe. The structure of the model is summarized in Fig. 14. And some hyper-parameters of model are summarized in Table 4.

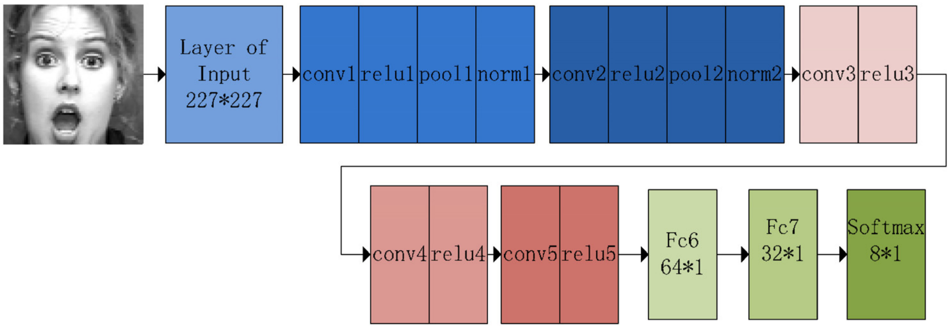

Fig. 14. The structure of our model for being trained on the CK+ dataset

Table 4. The training hyper-parameters

Parameters for solver	base_lr	Min_batch	lr_policy	gamma	Step size
Value	0.0001	100	step	0.1	1000
Parameters for solver	max_iter	momentum	weight_decay	Optimized algorithm	
Value	10000	0.9	0.00005	SGD	

4.3.3 Feature Visualization

When using the same method for the fine tuned model, we can attain the highly improved results shown in Fig. 15, Fig. 16 and Fig. 17.

These images in Fig. 15 are from the results of visualizing features on the third convolutional layer. The image in Fig. 15(b) reconstructed by all the neurons on the third convolutional layer takes on clear contour. The images in Fig. 15(c), Fig. 15(d), Fig. 15(e), Fig. 15(f) and Fig. 15(g) reconstructed by all the neurons of the 210th, 193th, 342th, 335th, 348th feature maps respectively show the significant features of face. The image in Fig. 15(h) reconstructed by top 1 activated neuron of all feature maps confirms that this convolutional layer is easier to learn eyes and mouth contours.

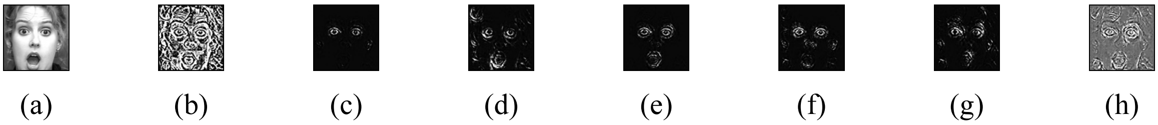

Fig. 15. The results of visualizing the third convolutional layers in the second transfer learning model

These images in Fig. 16 are from the results of visualizing features on the fourth convolutional layer.

The images are arranged in the same order as Fig. 15.

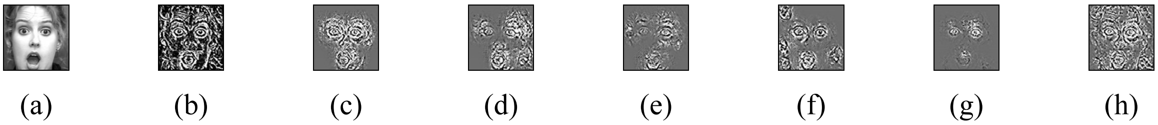

Fig. 16. The results of visualizing the fourth convolutional layers in the second transfer learning model

These images in Fig. 17 are from the results of visualizing features on the fifth convolutional layer.

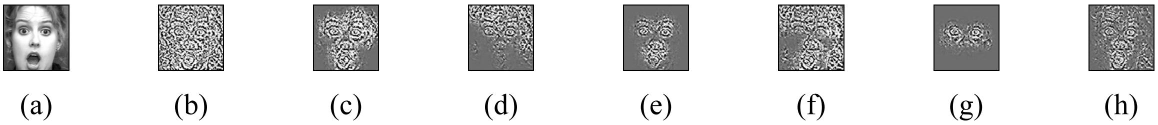

Fig. 17. The results of visualizing the fifth convolutional layers in the second transfer learning model

Observing results from the Fig. 15 to Fig. 17, we can make a conclusion that after being fine tuned on the CK+ dataset the third and fourth convolutional layers have learned the significant features of face powerfully while the fifth plays few positive roles in expression recognition. As a result, it leads us to modify this model in advance, as is shown in Fig. 18. In the experiment of expression recognition, we attempt to investigate whether the model with the fifth convolutional layer removed still performs well.

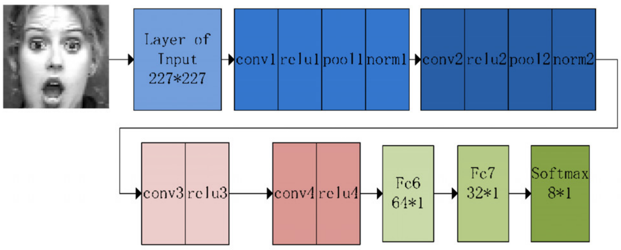

Fig. 18. The structure of our model for being trained on the CK+ dataset without the fifth convolutional layer

4.3.4 Experiment of Expressions Recognition

In the further modified model, only the two full connect layers need to be fine tuned on CK+ dataset again when reconnected to the fourth convolutional layer without the fifth convolutional layer. The hyper-parameters are the same as in Table 4. Upon completion of experiment, we compare our model without fifth convolutional layers with both state-of-the-art model and our model with fifth convolutional layers. The comparisons are summarized in Table 5.

Table 5. Comparisons among methods

Method	Average accuracy
Our model in Fig. 18	98.0%
Our model in Fig. 14	99.0%
AUDN [43]	93.7%
Zero-bias CNN+AD [9]	96.4%
CNN+SVM [44]	96.5%
CNN+SOFTMAX [44]	95.7%
CNN+Hypergraph [44]	97.3%

In the works [9, 44-45], authors come up with their creative methods to solve the 8 classes of expressions recognition on CK+, all of which refer to how to extract the low-level, medium-level and semantic features more effectively. For our model in Fig. 14, it outperforms AUDN, Zero-bias CNN+AD, CNN+svm, CNN+softmax and CNN+Hypergraph by 5.3%, 2.6%, 2.5%, 3.3% and 1.7%. Comparing the two models in Fig. 18 and Fig. 14, we can find there is no much big gap between the results. As is seen, it is obvious that the model in Fig. 14 is a little more complicated than in Fig. 18. So it is worthwhile weighing whether it is necessary to get a little more accuracy at the expense of more training and executing time for a practical system.

4.4 Experiment of Occlusion

To weigh whether our classifier is robust enough against interferences like the inputting image misses some regional information, we attempt to make some various occluding-different portions with a black box. The testing results are shown in Fig. 19, Fig. 20 and Table 6, where the pixel-level images are produced by feature visualization on the first and third convolutional layers and the accuracies are compared when the mouth, eyes and cheeks-foreheaded regions in the face of inputting images are occluded by rectangular black box respectively.

These images are from the results of visualizing features on the first convolutional layer. Fig. 19(a) and Fig. 19(b) are original inputting image and pixel-level feature-visualized image respectively. Fig. 19(d), Fig. 19(f), Fig. 19(h) and Fig. 19(j) are reconstructed by top 10 activated neurons of all feature maps corresponding to inputting image of Fig. 19(c), Fig. 19(e), Fig. 19(g), and Fig. 19(i), where the whole and left half mouth and eyes regions get black because the activations of the neurons mapped to those regions drop dramatically. That shows the first convolutional layer can genuinely learn significant features on face. While the cheeks and forehead which are supposed to be insignificant parts are occluded like (k), the pixel-level feature-visualized image in Fig. 19(l) is almost the same as Fig. 19(b). That means the layer genuinely learned no insignificant parts which play insignificant roles in recognizing expressions.

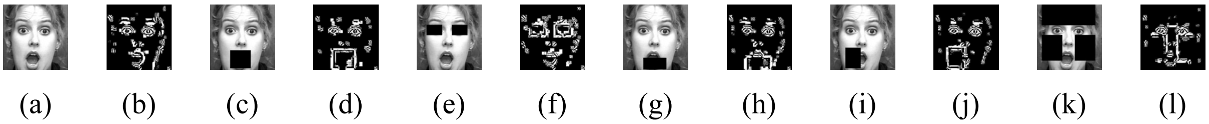

Fig. 19. The occluding results of visualizing features on the first convolutional layer

These images are from the results of visualizing features on the third convolutional layer. Fig. 20(d), Fig. 20(f), Fig. 20(h), Fig. 20(j) and Fig. 20(l) are reconstructed by top 1 activated neurons of all feature maps corresponding to inputting images in Fig. 20(c), Fig. 20(e), Fig. 20(i), Fig. 20(g) and Fig. 20(k) respectively. We can find the same conclusions as are discussed in Fig. 19.

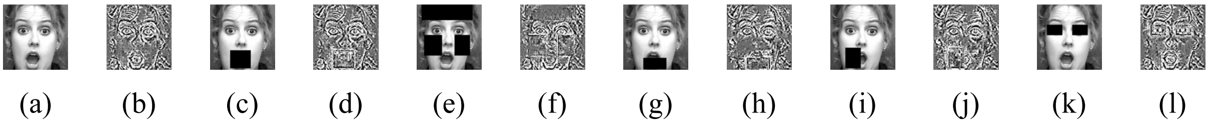

Fig. 20. The occluding results of visualizing features on the third convolutional layer

As checking the accuracies on the testing dataset of CK+ to quantify the effects on occluding different parts on face, we can find that occluding both cheeks and forehead takes little impacts on the accuracy of our classifier, which means that these regions play tiny roles in expressions recognition and the interferences locating in those regions does not affect the results much. But when the whole and half mouth are occluded, the accuracies get dropped dramatically. Especially, occluding both eyes and mouth directly results in failing to work, where the accuracy falls down to 9.5%. Otherwise we can also find the mouth is much superior to the eyes, even though the classifier is mainly dependent on both these two regions, which is the very way human observe parts of face to judge expressions. To some extent, the classifier can tolerate some interferences in the region neighboring to eyes while it would be in dysfunction when the mouth is occluded totally or partially.

Table 6. Testing results of occlusion on our model in Fig. 14

Occluding portions on face	Average accuracy
no occluding portions on face	99.0%
Occluding cheeks and forehead	96.5%
Occluding eyes	94 %
Occluding left half mouth	74.5%
Occluding bottom half mouth	73.0%
Occluding mouth	59.0%
Occluding mouth and eyes	9.5%

5 Conclusions

We introduce a framework containing a dual transfer learning and feature visualization for image-based expressions recognition on a small dataset. In the first transfer learning process, the results verify that the first convolutional layer of the original BVLC's model has learned low-level features effectively which can be shared with image-based expressions recognition. In the second transfer learning process, the learning knowledge zooms out to more narrow scope which focuses on attributions of face. Via being trained on CelebA dataset, the results demonstrate that the second convolutional layer can be used immediately in the second transfer learning model. In experiments of expression recognition, according to the results of feature visualization after fine tuning the model on CK+, we make an attempt to trim off the last convolutional layer to construct a little simpler network so that it can take less time to train and execute without reducing a lot of accuracies. The results indicate that both our two models perform better than the state-of-the-art methods. At last, we corroborate our model is robust against interferences of missing some information on face which are not critical parts for recognizing expressions.

In features visualization, we use the convolution and deconvolution algorithms in forward process and backward process respectively to reconstruct the targeting neurons in layer-wise mode into pixel-level image, which is observed and judged by eyes of persons to determine whether it is transferable or not. To some extent, the method of features visualization is subjected to experiences of researchers. So in the future work, it is indispensable to propose a quantitative model to evaluate the reconstructive image objectively and the CNN based on our proposed method should embody intensity of emotions based on the theory of emotional dimensions when the facial expressions are recognized, where CNN needs training to be a regression model. Additionally, the methods combining dual transfer learning and feature visualization not only can be applied in image-based facial expressions recognition but also be suitable for other fields of deep learning on small dataset.

Acknowledgements

This work was partially co-sponsored by the National Natural Science Foundation of China (Grants 61761166005), Ministry of Science and Technology, Taiwan (MOST 106-2218-E-032-003-MY3), and National Natural Science Foundation of Zhejiang (Grants LY20F020007), and the Ningbo Science Technology Plan projects (Grants 2019C50024, 2019C50081), and the Foundation of Zhejiang Educational Commission (Grants Y201738147), the Taizhou Science Technology Plan projects (Grants 1501KY61).

References

[1] D. McDuff, R. Kaliouby, T. Senechal, D. Demirdjian, R. Picard, Automatic measurement of ad preferences from facial responses gathered over the Internet, Image & Vision Computing 32(10)(2014) 630-640.

[2] N. Fragopanagos, J.G. Taylor, Emotion recognition in human-computer interaction, Neural Networks 18(4)(2005) 389-405.

[3] Y.-l. Tian, T. Kanade, J.F. Colin, Recognizing action units for facial expression analysis, IEEE Transactions on Pattern Analysis & Machine Intelligence 23(2)(2001) 97-115.

[4] Y. Tong, W. Liao, Q. Ji, Facial action unit recognition by exploiting their dynamic and semantic relationships, IEEE Transactions on Pattern Analysis & Machine Intelligence 29(10)(2007) 1683-1699.

[5] P. Ekman, W. Friesen, Facial Action Coding System: A Technique for the Measurement of Facial Movement, Consulting Psychologists Press, Palo Alto, 1978.

[6] J. Whitehill, C.W. Omlin, Haar features for FACS AU recognition, in: Proc. 2006 International Conference on Automatic Face and Gesture Recognition, 2006.

[7] G. Zhao, M. Pietikainen, Dynamic texture recognition using local binary patterns with an application to facial expressions, IEEE Transactions on Pattern Analysis & Machine Intelligence 29(6)(2007)915-928.

[8] G. Levi, T. Hassner, Emotion recognition in the wild via convolutional neural networks and mapped binary patterns, in: Proc. 2015 ACM on International Conference on Multimodal Interaction, 2015.

[9] P. Khorrami, T.L. Paine, T.S. Huang, Do deep neural networks learn facial action units when doing expression recognition, in: Proc. 2015 IEEE International Conference on Computer Vision, 2015.

[10] J. Jeon, J.-C. Park, Y.J. Jo, C.M. Nam, A real-time facial expression recognizer using deep neural network, in: Proc. 2016 International Conference on Ubiquitous Information Management and Communication, 2016.

[11] S.E. Kahou, C.J. Pal, Z. Wu, Combining modality specific deep neural networks for emotion recognition in video, in: Proc. 2013 ACM on International Conference on Multimodal Interaction, 2013.

[12] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: Proc. 2014 European Conference on Computer Vision, 2014.

[13] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, H. Lipson, Understanding neural networks through deep visualization, in: Proc. 2015 International Conference on Machine Learning on Deep Learning, 2015.

[14] P. Lucey, J.F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, I. Matthews, The extended Cohn-Kanade Dataset (CK+): a complete dataset for action unit and emotion-specified expression, in: Proc. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010.

[15] K. Yu, Z. Wang, L. Zhuo, J. Wang, Z. Chi, D. Feng, Learning realistic facial expressions from web images, Pattern Recognition 46(8)(2013) 2144-2155.

[16] L. Španić, Application of Microsoft Cognitive Services to emotion recognition in facial expressions, [dissertation] Croatia, Zagreb: Tehničko veleučilište u Zagrebu, 2017.

[17] D. McDuff, R. Kaliouby, T. Senechal, M. Amr, J.F. Cohn, R. Picard, Affectiva-MIT Facial Expression Dataset (AM-FED): naturalistic and spontaneous facial expressions collected, in: Proc. 2013 IEEE Conference on Computer Vision and Pattern Recognition, 2013.

[18] D. Ghimire, J. Lee, Geometric feature-based facial expression recognition in image sequences using multi-class AdaBoost and support vector machines, Sensors 13(2013) 7714-7734.

[19] S.L. Happy, A. George, A. Routray, A real time facial expression classification system using local binary patterns, in: Proc. 2012 International Conference on Intelligent Human Computer Interaction, 2012.

[20] D. Ghimire, S. Jeong, J. Lee, S.H. Park, Facial expression recognition based on local region specific features and support vector machines, Multimedia Tools and Applications 76(6)(2017) 7803-7821.

[21] M. Suk, B. Prabhakaran, Real-time mobile facial expression recognition system: a case study, in: Proc. 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[22] H. Soyel, H. Demirel, Facial expression recognition using 3D facial feature distances, in: Proc. 2007 International Conference Image Analysis and Recognition, 2007.

[23] H. Soyel, H. Demirel, 3D facial expression recognition with geometrically localized facial features, in: Proc. 2008 International Symposium on Computer and Information Sciences, 2008.

[24] C. Orrite, A. Gañán, G. Rogez, HOG-based decision tree for facial expression classification, in: Proc. 2009 Iberian Conference on Pattern Recognition and Image Analysis, 2009.

[25] M. Dahmane, J. Meunier, Continuous emotion recognition using Gabor energy filters, in: Proc. 2011 International Conference on Affective Computing and Intelligent Interaction, 2011.

[26] K. Sikka, T. Wu, J. Susskind, M. Bartlett, Exploring bag of words architectures in the facial expression domain, in: Proc. 2012 International Conference on Computer Vision, 2012.

[27] Y. Zhu, F.D.l. Torre, J.F. Cohn, Y.-J. Zhang, Dynamic cascades with bidirectional bootstrapping for action unit detection in spontaneous facial behavior, IEEE Transactions on Affective Computing 2(2)(2011)79-91.

[28] P. Liu, S. Han, Z. Meng, Y. Tong, Facial expression recognition via a boosted deep belief network, in: Proc. 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[29] Y. Bengio, Learning deep architectures for AI, Foundations and Trends in Machine Learning 2(1)(2009) 1-127.

[30] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: Proc. 2012 International Conference on Neural Information Processing Systems, 2012.

[31] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proc. 2015 International Conference on Learning Representations, 2015.

[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proc. 2015 IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[33] G. Larsson, M. Maire, G. Shakhnarovich, FractalNet: ultra-deep neural networks without residuals, in: Proc. 2017 International Conference on Learning Representations, 2017.

[34] R. Breuer, R. Kimmel, A deep learning perspective on the origin of facial expressions. <https://arxiv.org/abs/1705.01842>, 2017.

[35] H. Jung, S. Lee, J. Yim, S. Park, J. Kim, Joint fine-tuning in deep neural networks for facial expression recognition, in: Proc. 2015 IEEE International Conference on Computer Vision, 2015.

[36] K. Zhao, W.-S. Chu, H. Zhang, Deep region and multi-label learning for facial action unit detection, in: Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[37] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks, in: Proc. 2014 International Conference on Neural Information Processing Systems, 2014.

[38] Z. Yu, C. Zhang, Image based static facial expression recognition with multiple deep network learning, in: Proc. 2015 ACM on International Conference on Multimodal Interaction, 2015.

[39] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, in: Proc. 2014 ACM international conference on Multimedia, 2014.

[40] H.-W. Ng, V.D. Nguyen, V. Vonikakis, S. Winkler, Deep learning for emotion recognition on small datasets using transfer learning, in: Proc. ACM on International Conference on Multimodal Interaction, 2015.

[41] M. Peng, Z. Wu, Z. Zhang, T. Chen, From macro to micro expression recognition: deep learning on small datasets using transfer learning, in: Proc. 2018 IEEE International Conference on Automatic Face & Gesture Recognition, 2018.

[42] W. Zhi, Z. Chen, H.W.F. Yueng, Z. Lu, S.M. Zandavi, Y.Y. Chung, Layer removal for transfer learning with deep convolutional neural networks, in: Proc. 2017 International Conference on Neural Information Processing, 2017.

[43] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: Proc. 2015 IEEE International Conference on Computer Vision, 2015.

[44] M. Liu, S. Li, S. Shan, X. Chen, AU-inspired Deep Networks for Facial Expression Feature Learning, Neurocomputing, 159(2015) 126-136.

[45] Y. Huang, H. Lu, Deep learning driven hypergraph representation for image-based emotion recognition, in: Proc. 2016 ACM International Conference on Multimodal Interaction, 2016.