PhD Thesis Thesis:
Deep Learning Based Facial Computing
- Data, Algorithms and Applications

Wei Li
Department of Electrical Engineering
Grove School of Engineering
The City College of The City University of New York

Advisor: Professor Zhigang Zhu

May 29, 2017

Contents

Acknowledgements

I would like to express my special appreciation to my advisor Dr. Zhigang Zhu, Professor of Computer Science in Grove School of Engineering, The City College of New York, who has been a tremendous mentor to me. I would like to thank him for encouraging my research and for allowing me to grow as a researcher. His advice on both research as well as on my career have been priceless.

I would also like to thank Professor Jizhong Xiao and Professor Yingli Tian, both in the Department of Electrical Engineering at The City College of New York, Professor Jianting Zhang from the Department of Computer Science, and Dr. LijunYin, Professor of Computer Science at Binghamton University, The State University of New York, for serving as my committee members. I want to thank them for their brilliant comments and suggestions so that my Ph.D. research has been an enjoyable experience. I would especially like to thank the fellow Ph.D. students in the City College Visual Computing Lab and my friends in the Department of Electrical Engineering for discussing questions and ideas related to my research.

This work is supported by the National Science Foundation through Award #EFRI-1137172, and partially VentureWell (formerly NCIIA) through Award #10087-12. I would also like to thank IBM China Research Lab for the summer internship that enables the collection of the CIFE dataset. Special thanks to Ms. Celina M. Cavalluzzi, Director of Day Services, GoodWill, for her assistance in evaluating the game apps by individuals with Autism Spectrum Disorders and Professor Jianting Zhang for sharing his server for supporting my deep learning training.

Abstract

In this thesis, a complete task workflow of facial computing is introduced, which includes three compnents: data, algorithms and applications. Then the research focuses on two most important tasks in facial computing: facial expression recognition and face action unit (AU) detection, since the former is key indicators of people's emotion and the latter is the basic elements for more complicated facial tasks.

In the facial expression recognition part, we propose a recursive framework to recognize facial expressions from images in real scenes. Unlike traditional approaches that typically focus on developing and refining algorithms for improving recognition performance on an existing dataset, we integrate four important components in a recursive manner: facial dataset generation, facial expression recognition model building, and interactive interfaces for testing and new data collection, and finally a dataset evaluation and cleansing. To start with, we first create a candid-images-for-facial-expression (CIFE) dataset from Web images. We then apply a convolutional neural network (CNN) to CIFE and build a CNN model for web image expression classification. In order to increase the expression recognition accuracy, we also fine-tune the CNN model and thus obtain a better CNN facial expression recognition model. Based on the fine-tuned CNN model, we design a facial expression game engine and collect a new and more balanced dataset, GaMo. The images of this dataset are collected from various facial expressions that our game users make when playing the game. Finally, we run yet another recursive step – a self-evaluation of the quality of the data labeling and propose a self-cleansing mechanism for improve the quality of the data. We evaluate the GaMo and CIFE datasets and show that our recursive framework can help build a better facial expression model for dealing with real scene facial expression tasks.

In the AU detection part, we propose a deep learning based approach for facial action unit detection by enhancing and cropping regions of interest of face images. The approach is implemented by adding two novel nets (a.k.a. layers): the enhancing layers and the cropping layers, to a pretrained convolutional neural network (CNN) model. For the enhancing layers (noted as *E-Net*), we have designed an attention map based on facial landmark features and apply it to a pretrained neural network to conduct enhanced learning. For the cropping layers (noted as *C-Net*), we crop facial regions around the detected landmarks and design individual convolutional layers to learn deeper features for each facial region. We then combine the E-Net and the C-Net to construct a so-called Enhancing and Cropping Net (*EAC-Net*), which can learn both features enhancing and region cropping functions effectively. The EAC-Net integrates three important elements, *i.e.*, learning transfer, attention coding, and regions of interest processing, making our AU detection approach more efficient and more robust to facial position and orientation changes. Our approach shows a significant performance improvement over the state-of-the-art methods when tested on the BP4D and DISFA AU datasets. We have also studied the performance of the proposed EAC-Net under two very challenging conditions: (1) faces with partial occlusion and (2) faces with large head pose variations. Experimental

results show that (1) the EAC-Net learns facial AUs correlation effectively and predicts AUs reliably even with only half of a face being visible, especially for the lower half; (2) Our EAC-Net model also works well under very large head poses, which outperforms significantly a compared baseline approach. It further shows that the EAC-Net works much better without a face alignment than with face alignment as pre-processing, in terms of computational efficiency and AU detection accuracy.

To better address the problem of effective fusion of temporal information in AU detection, we propose a C-Net based region of interest (ROI) adaptation optimal LSTM-based temporal fusing approach. The optimal selection of multiple LSTM layers to form the best LSTM Net is carried out to best fuse temporal features. The temporal fusion can also be applied to facial expression recognition and other tasks.

1 Introduction

The research in computer vision is mostly about helping people in improving their quality of life. There are numerous computer vision topics targeting on human behavior and action understanding to providing useful information for human-to-human and human-to-computer interactions. By analyzing their actions, observing their facial reactions, and recognizing their gestures, what people are doing or how they are feeling can be understood. One of the most important topics among these works is facial computing. People pay a lot of attention to faces and all the information a face conveys. Good-looking faces are popular, confident and friendly faces are more welcome, whereas sad and depressed faces can obtain sympathy. For certain groups of people, reading faces are challenging and sometimes impossible, such as individuals with autistic spectrum disorder (ASD), or with visual impairment. Therefore a computing system that can read faces for them would be very useful. In this thesis, we call all these face related tasks as facial computing. The goal of our work is to investigate data, methods and applications that enable a facial computing system to read people's face and know people's feelings.

1.1 Data, algorithms and applications

To fully solve the problems in a facial computing task, we need to consider three important components: data, algorithms and applications.

Data are the basis for many computing vision problems. The popularity of convolutional neural networks is boosted with the ImageNet dataset, which covers thousands of objects in the world. Facial computing also relies on data. In our work, we will look into the problem of creating datasets useful for facial computing, by analyzing the problems that the current datasets have, and proposing our datasets or data collection frameworks for real world facial computing problems.

The second essential component is models or algorithms. We use algorithms to solve specific facial computing problems. Before we go into details of our algorithms we need define our problems first. *Facial computing* is to analyze people's facial attributes and expression, and obtain their inner facial expression or feeling. Based on what we want to achieve, there are several sub-problems in facial computing, such as facial expression recognition, arousal/valence detection and action unit detection.We would like to formally define them here.

(1). **Facial expression recognition** is to classify a facial image into one of the several (often 7) categories. The seven categories are defined as basic human facial expressions (Happiness, Anger, Disgust, Sadness, Surprise, Fear and Neutral), thus making this facial computing task into an image classification problem.

(2) **Facial arousal/valence detection** is to decide the levels of people's facial expressions. Arousal wants to know how positive an facial expression is whereas valence detection is to know bout the energy in one's facial expressions. This is a two-label regression problem.

(3) **Action Unit (AU) detection** wants to know the basic activeness of the basic facial action units. AU detection is a more basic problem comparing to facial expression recognition. There are usually more than 10 AUs being analyzed simultaneously. So, from the perspective of algorithmic processing, it's a mutli-label classification problem.

Among the three problems, facial expression recognition and AU detection are the two most studied topics. In our work, we target on the facial expression recognition and action unit recognition.

In terms of its significance and applications, facial computing has impact in all dimensions: scientifically, technologically, economically and socially. First, facial computing has a great scientific value in understanding humans capabilities in facial understanding. Second facial computing study could advance our knowledge in developing learning algorithms and models. Third, facial computing has a significant economical impact: being able to tell people's feelings can help vendors customize their personalized advertisement, obtain feedback from audience for all kinds of shows, achieving better understanding of students' learning progress in class, and helping detect deception, to name a few. Finally, for those who cannot read or who have difficulties in reading faces, such as people with ASD, facial computing can generate assistive tools for improving their social interaction thus improve the quality of their lives.

1.2 Overview of the thesis and contributions

This thesis include two closely related parts: facial expression recognition and action unit detection. Except that both parts use deep learning models and involve dataset selections, the latter can be building block of the former.

In the **facial expression recognition** part, we first build a candid image facial expression dataset - Candid Image Facial Expression (CIFE) – by parsing Web expression images from image search engines. The CNN-based deep learning approaches are then employed to train robust facial expression predictors, while fine-tuning approaches are also constructed to improve facial expression accuracy. To collect real scene images, we have designed a facial expression interaction game based on our deep learning model that was trained with the CIFE dataset. With users playing both the general and the customized versions of the face game, the correctly labeled facial expression images are selected and saved, which help us build the GaMo dataset. We also have run a self-evaluation of the quality of the data labeling and proposed a self-cleansing mechanism to improve the quality of the data. To prove the effectiveness of our framework, we compared GaMo and CIFE based their balancedness, recognition accuracy, the effectiveness of using strictly balanced subsets, the impact of data-cleaning, and feedback from human subjects. The experiments show that our framework can build a reliable facial expression predictor for real scenes. We have the following new contributions in facial expression recognition:

1. A recursive framework is proposed, which can recursively generate and automatically cleanse the new data, and update the deep learning model to have better performance in real scene facial expression recognition. In particular, we have a new recursive step to clean the collected data through a self-evaluation process.

2. A deeper CNN model is used by fine-tuning VGG - a 19-layer CNN structure proposed by the Visual Geometry Group [24]. The performance of the fine-tuned VGG networkis comparable to the state-of-the-art approaches on benchmarking facial expression-in-the-wild datasets, and performance comparison showed that it outperformed the results we generated using our initial CNN model and the fine-tuned AlexNet model we reported in our previous work.

3. We detail the design and evaluation of the game interface, which is only controlled by human facial expressions, and automatically collecting expression images while the players are playing the game. The new GaMo dataset is also analyzed leading to insights and new ideas for more balanced data collection with our recursive framework in the future.

4. The performance of facial expression recognition based on the two facial expression datasets is compared and analyzed: CIFE and GaMo, and their more balanced subsets using the new VGG model: CIFE is a web candid imaged based facial expression dataset; GaMo is an in Game-based facial expression dataset collected when our users played our facial expression game.

In the **Action Unit (AU) detection** part, we design an enhancing net (E-Net) to force the neural network to pay more attention to AU interest regions on face images. We have also proposed the cropping net (C-Net) to ensure that individual networks learn features in “aligned” facial areas. This makes our EAC-Net – the integration of the E-Net and C-Net – more robust to facial shifts and orientation differences. We also explored the LSTM-based temporal fusion approach, which boosted the AU detection performance significantly compared to static image-based approaches. In our AU detection approach, we have the following contributions:

1. The EAC-Net integrates three important elements - learning transfer, attention coding,

and regions of interest processing - which makes our AU detection approach more robust to face position and orientation changes.

2. No facial preprocessing such as normalization is required to apply to input images in our approach, which not only saves a significant amount of preprocessing time but also maintains the original facial expression representation unaltered. We have shown that our EAC approach has resulted in an excellent performance under large head poses, without a face alignment preprocessing step.

3. Although face landmarks are used in our C-Net and EAC-Net, they do not need to be very accurately located, i.e. the approach is robust to landmark detection errors.

4. The proposed EAC-Net is able to detect AUs even when faces in the input images are partially occluded with an impressive F1 score, close to the best performance of the state-of-the-art approaches on the BP4D dataset.

5. Multi-label learning is employed to integrate the outputs of those individual ROI cropping nets, which learns the inter-relationships of various AUs and acquires global features across sub-regions for AU detection. Multi-label and single AU based methods are compared. With additional AU correlations and richer global features, the multi-label learning approach shows slightly better performance.

6. A LSTM-based temporal fusion recurrent net (LSTM Net) is proposed to fuse static CNN features, which makes the AU predictions more accurate than with static images only.

1.3 Organization of the thesis

The thesis is organized as follows: We first introduce our thesis work in Chapter 1. After the introduction, the thesis is divided into 2 parts. In part I, we describe our work in facial expression recognition and in part II, details of AU detection work is provided.

Part I includes multiple aspects of work in facial expression recognition: After an overview of our facial expression work in Chapter 2, related work in facial expression recognition is discussed in Chapter 3. Our web collected facial expression dataset – CIFE is presented in Chapter 4. Then the CNN models for our facial expression recognition work are described in Chapter 5. To obtain more balanced and real scene data, a facial expression game is designed to collect the GaMo datase; these are discussed t in Chapter 6. Evaluation of the two dataset are conducted in Chapter 7. Finally in Chapter 8, we discussed how our facial expression work can help people in real applications.

As for the AU detection part, after an overview of this part in Chapter 9, the design considerations/ideas and related work are introduced in Chapter 10: an overview of design considerations is given in Section 10.1, and related work of AU detection is briefly discussed in Section 10.2. We then explain our proposed EAC-Net in Chapter 11 for static image AU detection and temporal fusion approach in Chapter 12. To evaluate our method, we test our approach with standard datasets in Chapter 13 and with our candid image dataset in Chapter 14. Possible AU detection based applications are mentioned in Chapter 15. And finally, we conclude the thesis and point our some future directions in Chapter 16.

Part I

Facial Expression Recognition: a Recursive Framework

2 Overview of the Recursive Framework

Detecting people's facial expressions has been an interesting research topic for more than 20 years. Facial expressions play an important role in many applications such as advertising, social interaction and assistive technology. Research in facial expression recognition mainly includes three parts: datasets, algorithms and real world interaction applications. In our approach, to improve the performance of facial expression recognition in real scenes, we propose a framework for integrating dataset construction, algorithm design and interaction implementation.

Even though facial expression research should include three integrated parts, most of the previous work mostly focused on one of the components. Consequently, algorithms or models designed for one or several datasets do not work well when dealing with real scene problems. With this in mind, we propose a recursive updating approach. Starting from a deep learning model trained from facial images collected from the Web, a facial expression game is designed for collecting new and more balanced data. Then the newly collected data are used to update the training model. The framework is illustrated in Figure 2.1. We start with a dataset with Candid Images for Facial Expression (CIFE) to build an initial facial expression model, which is then served as the game engine for a facial expression game, then when users play the game, facial images of the users are classified as different facial expressions by the model and automatically collected. This leads to a new and balanced dataset, named GaMo (standing for Game-based facial expression), which is cleansed with a self-evaluation mechnism and then used to update our facial expression model.

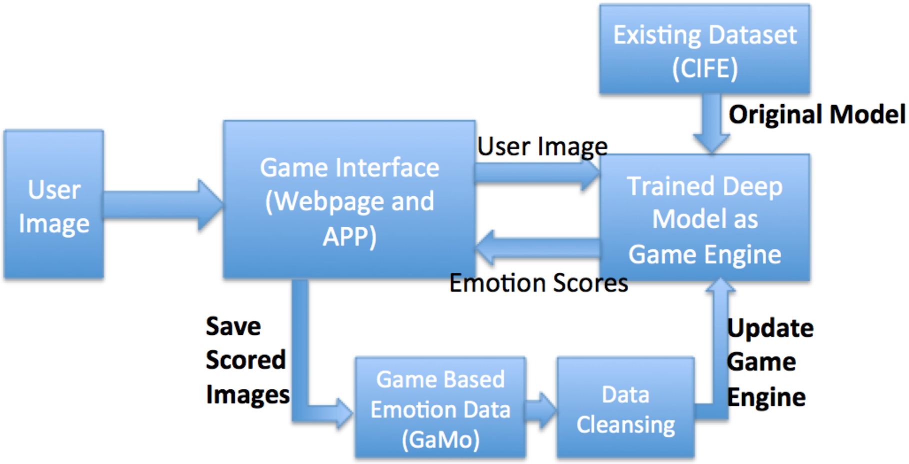


Figure 2.1: The proposed recursive framework

3 Facial Expression Recognition: Related Work

We already mentioned that current facial expression research mainly includes three major components: datasets, algorithms/models, and applications. Here we would like to give a review of each of the three components.

3.1 Datasets

Among the many datasets that have been provided by researchers for recognizing expressions from images, there are mainly two kinds of datasets. Datasets belonging to the first category are captured in laboratory. These include CK+, MMI and DISFA dataset [2, 3, 4, 5]. Usually subjects are invited to their labs and sit in lighting and position constrained environments. Good results can be achieved on these datasets but in real life scenarios, it's always hard to have good performance. Datasets in the second category are collected from existing media and social networks, such as Kaggle and EmotiW [7, 8, 10]. Using web search engines, one can easily obtain thousands of images but the datasets are usually not balanced. EmotiW is a video clip dataset for an expression recognition challenge, and the video samples are from Hollywood movies where the actors show different expressions. For the datasets collected from existing media, some of the expressions like Happy or Sad are easier to obtain, but for some expressions like Disgust or Fear, it's hard to find enough samples.

3.2 Algorithms and models

Although the existing datasets are generally not balanced, many interesting and promising approaches have been proposed for expression detection. Most existing facial expression recognition methods have focused on recognizing expressions of frontal faces, such as the images in CK+ [5]. Shan, et al [6] have proposed a LBP-based feature extractor combined with an SVM for classification. In the method proposed by Xiao, et al [11], instead of training one model for all expressions, separate models have been trained for each expression, which improve the overall performance. Wang, et al [12] modeled facial expressions as complex activities that consist of temporally overlapping sequences of face events. Then, an Interval Temporal Bayesian Network (ITBN) was used to capture the complex temporal information. Karan, et al [13] proposed a HMM-based approach to make use of consecutive frame information to achieve better expression recognition accuracy from video.

In the past few years, deep learning methods have been successfully used for face recognition and verification [62, 15]. Deep learning approaches are also used in many expression detection applications. Liu, et al [16] proposed a Boosted Deep Belief Network to perform feature learning, feature selection and classifier construction for expression recognition. Different DBN models for unsupervised feature learning in audio-visual expression recognition have been compared in the work done by Kim, et al [17]. Our early work [7] used CNNs on images collected from the Web. To prove the effectiveness of CNNs, we compared CNN-based facial expression performance on CK+ to the state of the art methods. Multimodal deep learning approaches have been applied to facial expression recognitions tasks. An example is Jung, et al's work [18] in which facial landmarks based shape information and image based appearance information are learned through a combined CNN network. The results show that deep learning based multimodal features act better than individual modalities or the use of traditional learning approaches. Automatically learned features have also been used for multimodal facial expression recognition on video clips [8].

3.3 Interactive applications

In the generation of ImageNet dataset [19], Amazon Mechanical Turk (AMT) is used to label all the training images. Workers are hired online and can remotely work on labeling the dataset. The ImageNet is a large scale dataset that aims to label 50 million images for object classification and without the help of online workers, the labeling would not be feasible. This inspired us to develop the idea of involving people in the data collection process through an online framework, preferable using games. There have been some efforts in using games to attract people to perform some image classification work. Luis, et al [20] designed an interactive system that attracted people to label images, Mourao et al [21] developed a facial engaging algorithm as the controller to play their Novoexpressions game, and a player engagement dataset was obtained and the relationship between the players' facial engagement and game scores were analyzed. But their goal is not to collect data. Expression games have also been used to entertain children with Autism Spectrum Disorder (ASD) and to help them perform facial expressions by mirroring their expressions to some cartoon characters [22]. Our online framework not only makes use of online crowdsourcing through games, but also has much lower cost than AMT. And since the numbers of various facial expressions can be controlled by the design of the games, the dataset can be guaranteed to be balanced.

4 CIFE: Dataset from the Web

Since we would like to develop facial expression approaches that can be used in real world scenes, we need to train and test the models on non-posed images, or candid images. Therefore we collect a Candid Image Facial Expression (CIFE) Dataset. We note that most of the facial expression images on the Web are randomly posed, and most of the expressions are natural. Therefore we use web crawling techniques to acquire candid expression images from the Web, and create our candid image facial expression dataset CIFE.

4.1 CIFE data collection

As we have mentioned, we define seven types of expressions: Happy, Anger, Disgust, Sad, Surprise, Fear and Neutral. Using related key words to the each of the 7 expressions in addition to the name of the expression (e.g., joy, cheer, smile for Happiness), we have collected a large number of images that belong to each of the seven expressions. We have used most of the image search engines, including Google, Baidu and Flickr. In our initial CIFE dataset [7], the number of samples of different expressions were: Anger (1785), Disgust (266), Fear (781), Happiness (3636), Neutral (644), Sadness(2485) and Surprise(997).

The images are from the web and most of them are not posed. However, the number of samples in different classes was highly unbalanced. Therefore, we have added some images to classes with fewer samples (for example Disgust and Fear) to balance the class sizes [8]. At the end, we obtained 14,756 images for 7 classes (after some manual post-filtering by humans). The total number for each facial expression in our revised CIFE dataset is listed in Table 4.1. This is the dataset we use in this thesis. In [8], the CNN model was one of the modules for video expression recoginition, but here we focus on facial expression recognition in single images. Figure 9.1 shows a few typical examples of faces with various poses.

4.2 CIFE data augmentation

Table 4.1: Sample numbers of the seven facial expressions in CIFE (Ang, Dis, Fea, Hap, Neu, Sad, Sur represent angry, disgust, fear, happy, sad, surprise, respectively).

Expr	Ang	Dis	Fea	Hap	Neu	Sad	Sur
Nums	1905	975	1381	3636	2381	2485	1993

Deep learning with CNNs always requires a very large number of training images in order to train a large number of parameters of the network for obtaining good classification results. Even though our CIFE dataset has 14,756 images for 7 classes, it is still insufficient

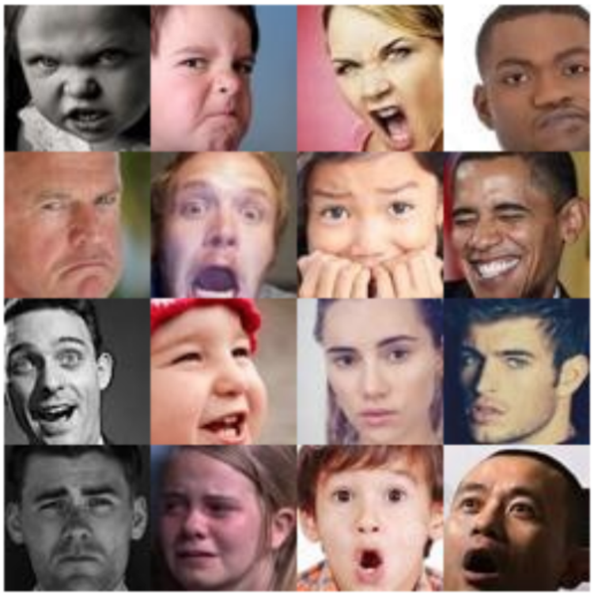


Figure 4.1: Images from CIFE

for training a deep CNN model. So before training the CNN model, we need to augment the dataset with various transformations to generate various small changes in appearances and poses. We applied five image appearance filters and six affine transform matrices. The five filters are disk, average, gaussian, unsharp and motion filters, and the six affine transforms are formalized by adding slight geometric transformations to the identity matrix, including a horizontal mirror image. Figure 4.2 shows an example of the facial image augmentation. By doing this augmentation, for each original image in the dataset, we can generate 30 (=5x6) samples, therefore the number of possible training samples would increase from 10330 be 309900, which is sufficient for training the deep learning model.

Data augmentation could also a very effective solution to some of the data problems. For example, there could be different image illumination/occlusion/pose situations in the original data. If we add various image transformations – in both appearance and geometry – to the original data, various situations can be simulated in the training dataset so that the obtained model will be more robust to all kinds of test data.

After data augmentation, we now have 309900 training images, and the model will be tested on 4,424 original testing images (30% of 14,756). Our goal is to classify all the images into 7 facial expression groups. To achieve our goal we design an CNN structure. In the following, we will describe our initial CNN model, the fine-tuning of two CNN structures -

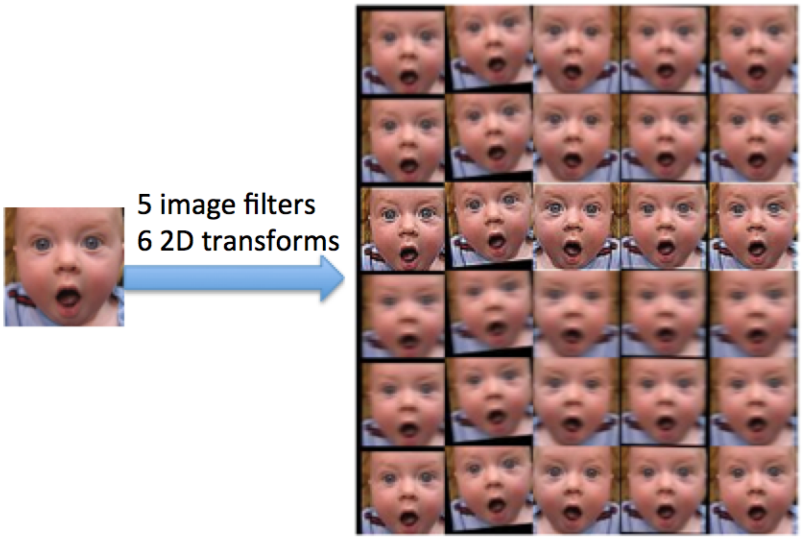


Figure 4.2: CIFE data augmentation

ALexNet [25] and VGG [24] , and report the comparison of their performance.

5 Algorithms: Evolvement of the CNN Structures

5.1 The initial CNN model

Our initial CNN model structure includes one input layer (the original image), three convolutional layers, and an output layer. This structure was arrived by trial and error with many experimental tests. The input color image size is 64x64, and the number of the output is 7. We set the convolutional filters size to be 3x3. We then varied the number of layers and the number of filter for each layer. After many rounds of tests, we finally found our most suitable "simple" structure with 3 convolutional layers, and the filter numbers for each layer to be 32, 32, 64, respectively. For each of the three convolutional layers, we add a 2:1 pooling layer to make the training data less redundant. The input 64x64 RGB image is then recognized as one of the 7 labeled classes. With this structure, we can easily know the numbers of the parameters to be around 184,000. Compared to the number of training images (309,900), the structure setting is also appropriate. Finally, we achieved a 65.2% accuracy on our test data. Even though the accuracy was not very high, the CNN-based facial expression showed its obvious performance advantage over tradition approaches such as the results using support vector machines (SVMs), 62.3% with the LBP Feature and 59.7% with the SIFT feature. Details of the results with the traditional approaches can be found in our previous work at [7]. We want to note here that we reported a much higher classification accuracy (81.5%) using a similar CNN model. That was because the highly unbalanced numbers of samples in the initial CIFE dataset used in [7]: there the recognition rates for disgust and fear classes were very low, for both the original dataset and the revised dataset, and hence adding new samples decreased the overall recognition statistics. We suspect that the reason for the low performance was that the three-layer structure is unable to learn the features deeply enough.

5.2 Fine-tuning AlexNet

To further improve the performance of facial expression recognition using CNN, we noted that learned general classification models can be used for specific classification problems [23]. Since some existing learned models are deeply trained on large scale datasets, image features thus learned can be better features for recognition of other classification tasks. Therefore we are curious to find out if this can help improve facial expression recognition. To try out our idea, we did experiments by fine-tuning AlexNet [25] and VGG [24] structures.

In the AlexNet structure, there are 1 input layer, 5 convolutional layers and 3 fully connected layers, leading to 60 million parameters in total. Our first guess was that training the AlexNet on our CIFE dataset would results in better classification accuracy. The only

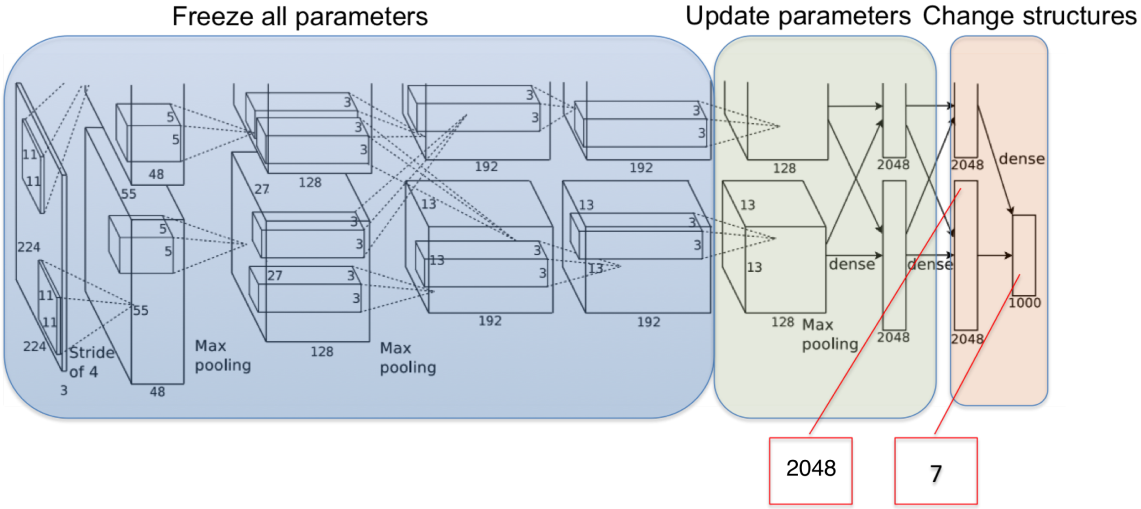


Figure 5.1: Fine-tuning AlexNet structure for facial expression recognition

problem was the need for a larger number of images, as the ImageNet requires millions of images during training.

Therefore, we instead propose a CNN fine-tuning method to train a deeper model based on AlexNet. The rationale is that although our task is different from the ImageNet, which focuses on object classification, similar low level filters could be used in expression recognition. Based on this hypothesis, we can use the AlexNet and utilize our relatively 'small' dataset to update and fine-tune parts of its parameters for adapting it to expression recognition.

As shown in Figure 9.2, the parameters of the convolutional layers 1 through 4 are not changed. Our new CIFE dataset is used to update the parameters of the convolutional layer 5 and the first fully connect layer, without changing their structures. In the original AlexNet, the number of units of the second fully connected layer and the third layer are 4096 and 1000 (classes) respectively. Since the number of classes in our dataset is just 7, we needed to change the structure in these two layers. We reduced the number of neurons in the penultimate layer to 2048, and the third fully connected layer to 7. The classification accuracy by using this model is 73.5% on the revised CIFE dataset, which shows that the fine-tuning leads to a much better performance than our first attempt of using a three-layer CNN structure, a 8.3% improvement. This was the model we used in collecting the GaMo facial expression dataset, when only this model was available. With this decent accuracy, a system that uses such a facial engine can lead to a good chance to obtain the right prediction in human computer facial interaction to encourage users to play the interaction game, which will be described in Chapter 6.

5.3 Fine-tuning VGG

Compared to AlexNet, VGG is a much deeper network. After the GaMo data collection, we also investigated if using a fine-tuned VGG model can improve the facial expression recognition performance. We first tested the fine-tuned VGG model on the revised CIFE dataset for comparing with the results with the fine-tuned AlexNet model. There are 19 learning layers in total, with 138 million parameters . VGG layers have some similar structure as AlexNet. They both have convolutional parts and fully connected parts. For each convolutional layer in AlexNet, VGG replaces it with 2-4 convolutional layers. Deeper networks lead to better representation of the input images: in the ImageNet challenge, VGG yielded a 6.8% top-5 error compared to AlexNet's 16.4%. We applied a similar fine-tuning approach to the VGG net as we did to AlexNet. By fintuning on existing VGG model with the revised CIFE dataset, we finally achieved a 76.3% accuracy, which is a 2.8% improvement over the fine-tuned AlexNet, and 11.1% over our initial CNN model. Therefore in Chapter 7, we will show results using the fine-tuned VGG structure for facial expression recognition with CIFE and GaMo datasets and their sub-sets.

Through finetuning the ImageNet models, we obtained improved facial expression recognition results. It indicates that the models for general image classification can share convolutional filters with specific purpose image classification tasks, such as facial expression recognition. In this way the fine-tuning leads to more robust models for non-posed image facial expression prediction.

Note that this model has been used to provide reliable CNN-based features in our work in participating at the Emotiw 2015 challenge on bench-marking ?facial expression recognition in the wild? datasets. This proves that the finetuning approach is comparable with the state of the art methods in solving the facial expression recognition- in-the-wild problem [8]. In our previous work, we show that the finetuning CNN feature is the most effctive feature among the three multimodal features we used –a LBP-TOP-based video feature, an openEAR energy/spectral-based audio feature, and the CNN based deep image feature, enabling us to achieve a rank 5 among 73 teams.

6 GoMo: Game Designs and Datasets

We already showed that deep learning can lead to high accuracy facial expression recognition. While the candid images are most likely randomly posed, but they are still different from images from real scenario interaction. More "real" data would be facial expression images of people collected without any constraints. If people can show this kind of "real" facial expression, we can use our facial expression model to select corresponding images and construct a real scene facial expression dataset. The selected images may be useful for building a real scene expression model. For this purpose, we decided to design a game interface that invites people to show their facial expressions, willingly, while playing a game.

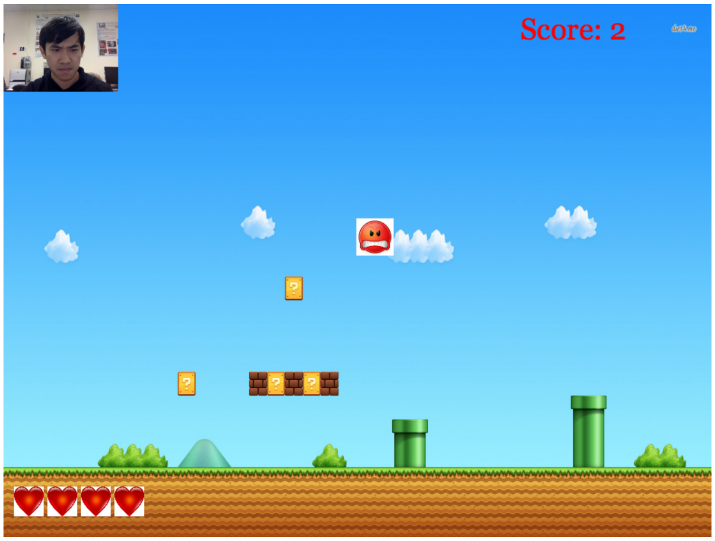


Figure 6.1: Design of the facial expression game scene

6.1 Game design

Since we would like game users to show real facial expressions yet remain engaged throughout the game, we had to design the expression game in a way that it is straightforward and interesting as much as possible. After performing research on the popular web games, we found that the Tower Defense games is the style that fits our task the best. The logic of the Tower Defense games is always very simple: the player needs to build a defense system against the intruders. In our application, we would like the user to act as a defender against an "expression target". An example of the basic game scene is shown in Figure 11.3. The live video of the user is shown on the top-left corner of the screen so he can always see his expression. A randomly picked expression target as shown in Figure 6.2 (a facial expression

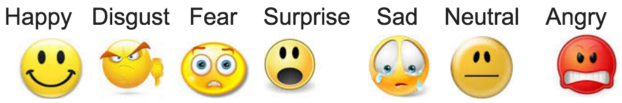


Figure 6.2: Facial icons representing the 7 basic expressions

icon) will enter the screen as a bomb dropped from above and the user has to protect the village by making the bomb disappear before it reaches the ground. Some sound effects are also added to make user more engaged. The bomb would disappear if the user makes a facial expression that correctly matches the displayed expression (the bomb), as judged by the CNN-based expression detector using the fine-tuned AlexNet model which was the available model to us when we collected the new data. The score as shown on the top-right of the screen will be increased.

6.2 General version and customized version

Here we would like to provide some technical details of our general game design. The expression game web interface accesses the camera on the user's machine and displays the video on the top-left corner of the screen. Then the game interface captures images of the user's face, and then sends face images to our server. The CNN model we trained by fine-tuning AlexNet analyzes each image and generates a probability vector for the seven expressions and sends it back to the game webpage. The reason for processing face images at a server is due to the high computational requirements by the CNN model. After the probability vector of the seven expressions is sent back to the game interface, it compares this feedback with the expression target ID that has been displayed on the screen as a bomb and informs the server to save the image if it matches the icon. Since the target facial expression that the user needs to make is defined by our system, we not only know the label and have high confidence that facial expression's label is correct, but also can make the dataset more balanced based on our needs.

The frame rates of typical webcameras are usually 20-60 Hz. We do not need such high frame rates of image capturing for two main reasons: 1) From the computational resource's point of view, the server will have a huge workload if we run the expression recognition on every single image since the CNN computation is time-consuming with hundreds of millions of parameters . 2) There is no need to know the user's expression frame by frame. Giving the user some time delay to prepare their expression may actually result in better image quality and as a consequence, a much better dataset. For these two reasons, we design the game in such a way that it only sends one image per second to the server. It takes only around 200 ms for the server to generate the results for every single image, which makes the game run very smoothly. We also set the number of initial game lives to be five and generate the expression targets randomly, with equal probabilities to all the seven expressions, which theoretically result in a balanced dataset.

The game is implemented using Javascript and HTML. The backend is hosted using Ruby on Rails. During the game, expression icons will drop from the top of the screen, and the user's face is also shown in the left top corner of the screen for the user to check her/his facial expression. If the user is able to match the expression before the icon hits the ground, she/he will receive 1 point. The score will change as the user gains or loses points. When all the game lives are used, a "Game Over" sign will be shown up, together with the total score gained by the user, and then a "Replay" button.

After making the game available to a small group and collecting data from several users who tried it, we realized that the collected dataset is not ideal, specifically for two main reasons: 1) Sometimes it is hard for users to correctly imitate the exact expressions by just looking to the icons; 2) Our expression detector sometimes is not able to correctly determine whether the subject is making the right face or not. This makes it hard for the players to achieve high scores and as a result, the collected data becomes imbalanced.

We were able to provide two solutions to solve the above problems. One is to change the expression recognition to an expression verification task. This makes the classification task much easier. Since we know the "ground truth" or the target label for an icon being displayed in the game, we only need to check if the probability of this specific expression reached a predefined threshold. Each expression needs to have its own threshold since some expressions are harder to mimic through facial expressions and have higher variety among different users. This will help the users achieve higher scores and also include a broad range of correctly labeled facial expressions for each expression in the dataset.

Another solution is to create an individual model for each player based on the CNN extracted features. The user will be compared with her/his individual expression templates instead of the general CNN model. The Deepface work [62] has proved that the CNNs is not only able to directly perform image classification, but can also extract robust features from the images. Thus we extract the features from the CNN for each individual user and then these features are saved as templates for that specific user. This makes the game customized for each player and the user can gain higher scores and is encouraged to play more.

As a part of the solutions proposed above, we designed a user registration page as shown in Figure 6.3. The registration page is divided into 8 subareas. The first subarea shows the current video stream. The other seven subareas display the seven registered expression templates. To save each template image, the user can click on the corresponding subarea

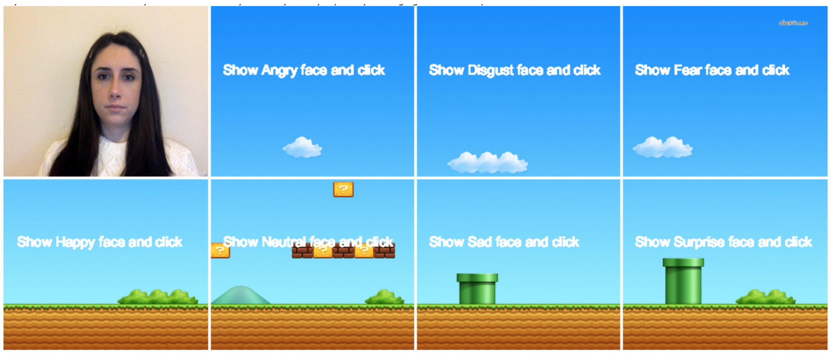

Show Angry face and click
Show Disgust face and click
Show Fear face and click
Show Happy face and click
Show Neutral face and click
Show Sad face and click
Show Surprise face and click

Figure 6.3: Registration page for building customized facial templates: initial page. The first image is the live webcam view, and the rest seven are places for showing the templates of the seven expressions.

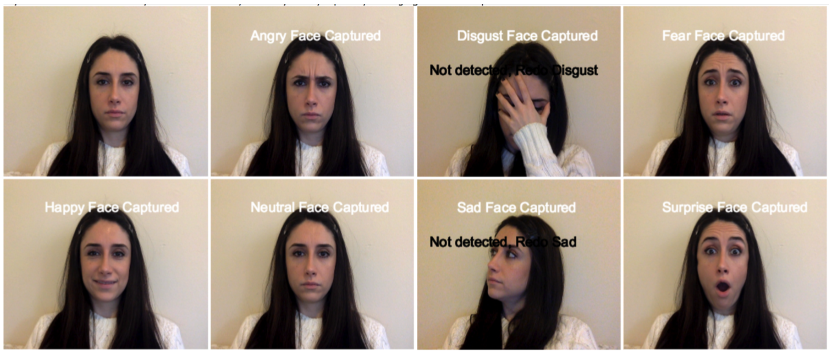


Figure 6.4: Faces of the seven expression after the user click "Send All". In this example, two of the images are not qualified so the user has to recapture these two images.

while imitating the correct facial expression. This process can be repeated several times until the user is happy with the saved image. Once all seven expressions are registered, the user can click the "Send All" button to send the expression templates to the server, where the system will detect the face area in the images and use the CNN model to extract expression features for the user. If the face cannot be detected, an error message will be sent back to the user, and she/he is then asked to recapture the image for the specific expression that has caused the error, as shown in Figure 13.1.

When all the template features are saved in the server, the user will be directed to the customized game scene. While the game is being played, the server will extract features for the image that is being sent at the moment and compare these features to the saved facial expression templates. We use L2 distance to select the nearest result and send it back as the detected facial expression. Since the features are robust and the user is always compared with her/his own model, the user will potentially achieve a higher score. We call this version of the game the ”customized version”, as opposed to the previous ”general version”.

6.3 GaMo: a New Facial Expression Dataset

Within one month of the release of the two game test versions to the college students of our department, more than a hundred users played the general version and 74 users tried the customized version. All the users that we collected data from have signed the consent form of our IRB approval. We obtained 15455 images in total during this time period and generated the GaMo (game based expression) Dataset. Compared to some deep learning datasets, the size is still not big enough, but our game can run at any time, so we can obtain a much bigger dataset when the game reaches more people. The dataset is available by contacting the authors.

One concern about our dataset might be the use of a trained model to obtain more expression data: Will this recognition/ verification model only create expression data that are similar to our existing data samples and make the dataset less diverse? Will the use of a pre-trained model affect the quality of the data labeling? We arrived at two observations:

1) The game interaction and incentive can increase the tendency of the users to express more accurate facial expressions for data collection. We found that in playing our game, users would rather achieve a higher score compared to the others. Thus they try to show the correct facial expression as well as they can. This mechanism can automatically help us avoid too many wrong labeled data with manual labeling, since the data we collected has been double-checked: by both our model and our users themselves.

2) In the customized game mode, user are comparing with their own facial expression templates. We checked all the templates of all the users and they were all correct. Different people have different facial expression patterns. By collecting templates and in-game facial expression images of different people, we can have diverse facial expression images.

We would like to note here that no manual cleanups for the images and labels have been done in the initial GaMo dataset; all the images are used in our first evaluation in Chapter 7. By randomly checking the dataset, we have not found any labels that are very off the real expressions. The distribution of the dataset is shown in Table 7.1. Compared to the CIFE dataset, GaMo is more balanced, which hopefully will result in a much more reliable facial expression detector. In conclusion, the data collection is automatic, of high quality and more balanced. Later on in Chapter 7, we will also discuss if an automatic data quality evaluation and cleanup could lead to better performance.

7 CIFE and GaMo Evaluations

Based on our proposed framework, we applied deep learning and finetuning to web collected images CIFE and obtain a facial expression recognition model - the fine-tuned AlexNet, then we used the model to host the facial expression game to collect the new GaMo data. We hope the GaMo dataset can be used to recursively finetune our CNN models. To prove this, we first need show that the GaMo data can actually improve the real scene facial expression recognition.

To determine the usefulness of the GaMo dataset, we performed the following experiments. First, we trained a new CNN model with GaMo by finetuning the previous AlexNet model that has been used as our game engine, which was trained on CIFE . To compare GaMo with CIFE, we ran both a self evaluation and a cross evaluation with the two CNN models: the GaMo CNN model and the CIFE CNN model.

In our earlier work [9], we have shown that the model trained with the more balanced GaMo dataset produced more robust results, especially for those classes that were underrepresented in the CIFE dataset. Further the GaMo model can be applied to the CIFE dataset with a decent performance, but not the other way around.

In order to see if these observations are consistent with more complicated and better performed models, we then used the fine-tuned deeper VGG models - the best models among the ones we have developed and used. The VGG models are trained with the CIFE and GaMo datasets, respectively, and performed the same experiments as in our earlier work.

We noted that due to the game engine we used which was based on the unbalanced CIFE dataset, we still had fewer samples in some categories, thus the new GaMo dataset was not completely balanced. Therefore we also ran an experiment to see if we just using more balanced sub-sets from both CIFE and GaMo will have large changes to the recognition results. Finally, we designed a small user study to find out if the dataset can actually improve the game engine and game experience. For this purpose, the users played the general version of the game hosted by the two new CNN models.

7.1 Comparison of CIFE and GaMo

Table 7.1 show the statistics of GaMo and CIFE datasets. For the CIFE dataset, as we mentioned before, the images are collected by searching from web engines using key words. We also went through the dataset to remove all the images that are not meaningful facial expressions. To some extent, the numbers of samples in the seven facial expression categories reflect the distribution of facial images numbers online. We can clearly see the imbalance of the sample numbers for different facial expression categories, and it's hard for us to balance

Table 7.1: Comparison of expression sample numbers in CIFE and GaMo

Dataset	CIFE	GaMo
Angry	1905	1945
Disgust	975	1838
Fear	1381	1586
Happy	3636	3185
Neutral	2381	2741
Sad	2485	1898
Surprise	1993	2262

Table 7.2: Average accuracies of self and cross evaluation of CIFE and GaMo models

	CIFE	GaMo	CIFE cross	GaMo cross
Average	0.76	0.75	0.31	0.64

it since if we use the minimal number 975 for Disgust, the numbers of samples would be too small. We can also see that the sample numbers of each facial expression from GaMo are more balanced. Although we noticed some of the expression numbers are also smaller than others like Fear and Disgust, it's easy for us to make it more balanced. When we design the game, we make all expressions show up at the same probability, but due to different ability of the expression prediction using the game engine trained with the unbalanced CIFE dataset, the GaMo dataset is not completely balanced. The good news is that, since we already have known the different accuracy in predicting each facial expression, in the future data collection with our recursive framework, we can change the show-up probabilities of of facial expression targets to control the final data distribution. This is our ongoing work.

7.2 Comparison of CNN models with CIFE and GaMo

To compare the two models, we test the overall accuracy in recognizing all the seven expressions (the average accuracy in Table 7.2) as well as the accuracy of each individual expression within its own sub-dataset (Angry, Disgust, Fear, Happy, Neutral, Sad, and Surprise), as listed also in Table 7.2, and in Table 7.3 to Table 7.6 . This would give us a good sense on the usefulness of GaMo dataset. Furthermore, to compare the performance of the two CNN models based on the VGG structure, we perform a cross evaluation: the model trained on CIFE is tested on images from GaMo and vice versa.

The confusion matrices of these four experiments are listed in Table 7.3 to Table 7.6 . Looking into the self evaluation results, we can see that the model trained on GaMo has a much more balanced distribution on expression classification on all the seven expressions.

Table 7.3: Self evaluation confusion matrix of CIFE

	Ang	Dis	Fea	Hap	Neu	Sad	Sur
Ang	0.81	0.03	0.02	0.01	0.03	0.06	0.03
Dis	0.07	0.53	0.06	0.03	0.19	0.03	0.06
Fea	0.04	0.02	0.62	0.02	0.04	0.07	0.2
Hap	0.02	0.02	0.001	0.85	0.02	0.05	0.02
Neu	0.05	0.09	0.02	0.02	0.70	0.04	0.08
Sad	0.07	0.01	0.01	0.03	0.04	0.82	0.01
Sur	0.01	0.01	0.08	0.02	0.07	0.01	0.78

Table 7.4: Self evaluation confusion matrix of GaMo

	Ang	Dis	Fea	Hap	Neu	Sad	Sur
Ang	0.62	0.07	0.02	0.04	0.11	0.08	0.04
Dis	0.06	0.69	0.03	0.06	0.05	0.08	0.01
Fea	0.02	0.05	0.62	0.05	0.11	0.02	0.1
Hap	0.02	0.02	0.01	0.81	0.04	0.05	0.02
Neu	0.01	0.02	0.02	0.03	0.85	0.02	0.02
Sad	0.02	0.05	0.02	0.05	0.04	0.77	0.02
Sur	0.02	0.02	0.06	0.04	0.04	0.01	0.79

Table 7.5: Cross evaluation confusion matrix of CIFE

	Ang	Dis	Fea	Hap	Neu	Sad	Sur
Ang	0.11	0.06	0.03	0.02	0.48	0.0	0.29
Dis	0.02	0.18	0.01	0.02	0.50	0.01	0.25
Fea	0.01	0.05	0.04	0.02	0.26	0.0	0.6
Hap	0.01	0.01	0.01	0.02	0.34	0.01	0.53
Neu	0.01	0.02	0.01	0.01	0.85	0.0	0.11
Sad	0.002	0.06	0.008	0.008	0.71	0.01	0.19
Sur	0.004	0.01	0.03	0.017	0.13	0.04	0.79

Table 7.6: Cross evaluation confusion matrix of GaMo

	Ang	Dis	Fea	Hap	Neu	Sad	Sur
Ang	0.71	0.02	0.01	0.01	0.01	0.13	0.03
Dis	0.13	0.28	0.01	0.16	0.01	0.36	0.02
Fea	0.01	0.02	0.44	0.14	0.02	0.11	0.18
Hap	0.02	0.01	0.001	0.91	0.01	0.04	0.01
Neu	0.12	0.05	0.02	0.14	0.24	0.39	0.03
Sad	0.07	0.01	0.01	0.08	0.02	0.82	0.01
Sur	0.04	0.01	0.11	0.17	0.02	0.05	0.58

Even though the average performance of the CNN model on the CIFE is slightly higher than that on GaMo, the numbers are misleading since the higher average accuracy of the CIFE-trained CNN model is due to the much larger numbers of samples in both Happy and Sad classes, which apparently also have much higher accuracy than others. In comparison, the performance in recognizing Disgust and Fear is much higher using GaMo than using CIFE.

The results of the cross dataset tests are even more interesting. The model trained on CIFE has a very low performance when tested on the GaMo dataset, the confusion matrix shows that many images are classed to neutral. We have observed that the difference between the images is significant among the two datasets. Our observations indicate that the expressions in the CIFE dataset tend to be more exaggerated and thus easier to be identified, as it are shown in Figure 6, while the GaMo dataset is more realistic to real life, as it is obtained from ordinary users with a high amount of varieties in imitating facial expressions while playing the game. As an example, Figure 7.1 shows two users who played the game. The first player shows more explicit expressions while the second player's expressions tend to be more implicit. This makes it hard for the model trained on CIFE to classify the images from GaMo. The CIFE model almost completely fails in recognizing Angry, Disgust and Happy in GaMo. We believe the reason is that these three expressions in the CIFE dataset, whether they have fewer or more samples, are much more highly exaggerated than those in the GaMo dataset. On the other hand, when the model trained on GaMo is cross-tested on CIFE, the performance is surprisingly good, even though the performance cannot beat that on the self-test. The reason is that the model is further fine-tuned on a larger, more inclusive and more balanced dataset. The GaMo model does reasonably well on all the three expressions failed by the CIFE model. In addition, if subtle expressions (as in the GaMo dataset) can be recognized, the exaggerated ones (as in CIFE) are not difficult to detect. As an example, the Happy faces in CIFE can be much more easily recognized (with a 91% accuracy) using the GaMo model.

Here we also want to note that the performance using the VGG structure is much better than using the AlexNet; interested readers please compare the results in Table 7.2 with the results in our previous work [9]. Nevertheless, the performance comparison observations between the CIFE and GaMo datasets are consistent from the AlexNet to VGG structure.

7.3 Comparison with more "balanced" sub-datasets

In most facial expression datasets, the sample numbers for different expression are imbalanced. The training process favors the class which has more samples to get higher accuracy. But this will weaken the model's ability to recognize the facial expression with less samples. In reality, this will not be a good interactive experience if the facial expression model is unable to recognize some less frequent facial expressions. So if we want to build a model that can

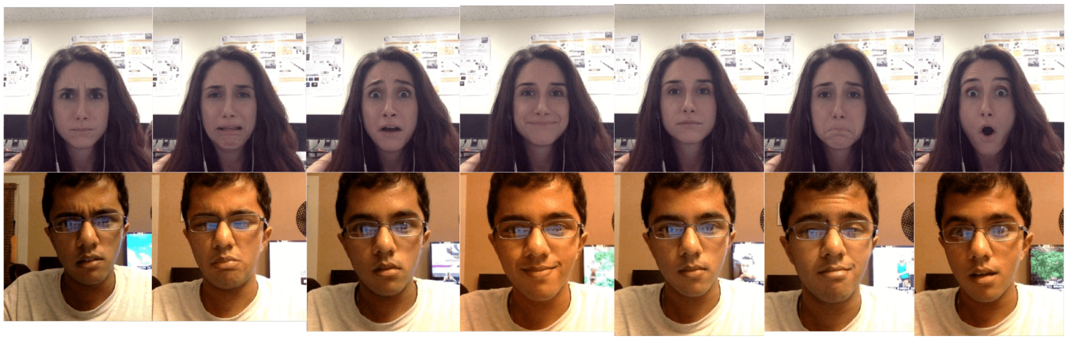


Figure 7.1: Comparison of individual template images of two users from GaMo

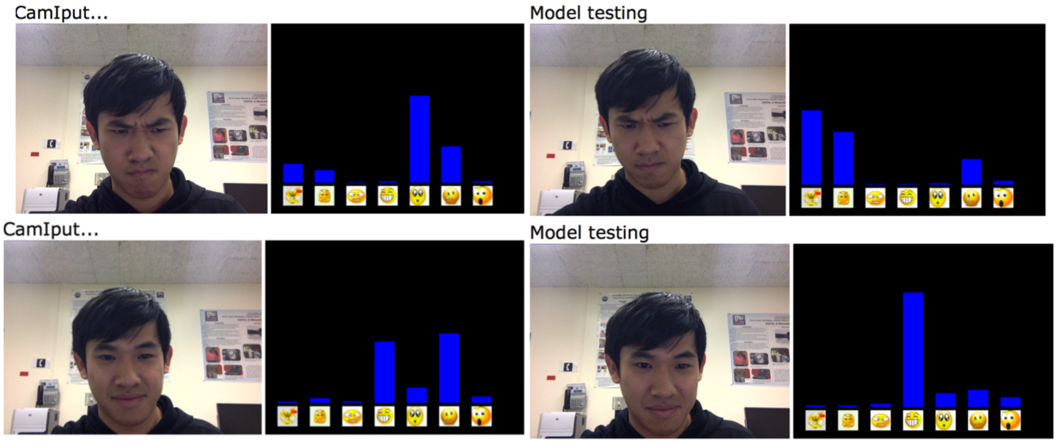


Figure 7.2: Subtle facial expression recognition by CIFE and GaMo models (left two are from the CIFE model and right two are from the GaMo model). Each histogram shows the probability distribution of the seven facial expressions for each facial image. The order of the expressions is Angry, Disgust, Fear, Happy, Neutral and Surprise. For some subtle expressions, only the GaMo model works well.

Table 7.7: Self evaluation confusion matrix of sub-balanced CIFE

	Ang	Dis	Fea	Hap	Neu	Sad	Sur
Ang	0.71	0.04	0.05	0.06	0.04	0.19	0.03
Dis	0.06	0.55	0.06	0.03	0.19	0.03	0.06
Fea	0.04	0.02	0.64	0.02	0.04	0.05	0.15
Hap	0.05	0.02	0.04	0.73	0.03	0.07	0.03
Neu	0.05	0.14	0.05	0.02	0.60	0.04	0.08
Sad	0.11	0.04	0.04	0.04	0.05	0.68	0.01
Sur	0.03	0.03	0.08	0.02	0.07	0.01	0.67

recognize all the facial expressions with equal accuracy, the best way is to create a training dataset with similar samples. In this case, for dataset CIFE, we will only have 4781=683x7 (683 is the number of Disgust expression samples in the CIFE train set, 70% of the total) images in total, which may not be sufficient for training a well-performed deep learning model. We use the subset of the CIFE dataset, which is balanced and run the same deep learning training as the full CIFE. We augment the 683 images of each facial expression and then finetune the VGG model. The final prediction result on CIFE is shown in Table 7.7. As predicted, by comparing Table 7.3 and Table 7.7, overall the performance is lower than using the full CIFE dataset, except the least frequent expressions: Disgust and Fear: The average recognition rate drops by 9%. But for the GaMo dataset, we still have over 10K images in the subset of balanced data, and the training data set has over 7770 images (1586x70%x7) . The performance result using the balanced GaMo subset is shown in Table 7.8. Compared to the result in Table 7.4, the overall performance improves by 6.4%. As a matter of fact, the recognition rates for all the categories increase; those with lower numbers of samples in the original GaMo datasets (Fear, Disgust and Sad) increase significantly, by more than 10%.

By comparing the two approaches in collecting facial expression images: searching from the Web, and harvesting from game users, we have some important notes. First, it's almost impossible for us to get more facial images for CIFE as we already have searched most of the image search engines in order to obtain high quality images. While for GaMo, as long as our game is running, we can have more and more balanced expression data. Second, even for the current version of GaMo, we retrained the deep learning model with the balanced subset of the GaMo dataset and by testing on the same original GaMo testing data, we see the performance has increased significantly.

In the balanced CIFE subset, due to fewer training data than the full CIFE, the performance for Angry and Happy dropped dramatically but the accuracy for Disgust and Fear doesn't improve much. While for the GaMo dataset, since each facial expression still has

Table 7.8: Self evaluation confusion matrix of sub-balanced GaMo

	Ang	Dis	Fea	Hap	Neu	Sad	Sur
Ang	0.68	0.08	0.08	0.02	0.06	0.03	0.02
Dis	0.02	0.83	0.03	0.03	0.02	0.03	0.03
Fea	0.01	0.01	0.85	0.02	0.04	0.05	0.05
Hap	0.02	0.03	0.02	0.84	0.03	0.03	0.02
Neu	0.02	0.03	0.04	0.02	0.86	0.04	0.02
Sad	0.02	0.03	0.03	0.02	0.02	0.87	0.01
Sur	0.02	0.02	0.06	0.02	0.02	0.01	0.86

more than 1586 images, the balanced subset of GaMo is still a good dataset for training. The balanced GaMo produced a better facial expression model than the full GaMo. For the less representative facial expressions like Disgust, Fear and Sad, the improvement is huge. The reason for this is that with equal consideration of all facial expressions during the training process, all expressions' deep features can be learned correctly, and if the test data can be well represented by the training data, we can achieve very good results. So, with our framework, we have a better chance to be able to obtain a robust expression predictor on all facial categories.

7.4 Dataset cleaning via sample evaluation

The CIFE dataset was generated by using image search engines with keywords related to the seven facial expression classes. On the other hand the GaMo dataset was collected by using a facial expression recognition engine trained with the CIFE dataset for checking users facial expression matching. Although we have manually run a proof-check on the quality of CIFE, and have found that most human subjects (game users) in our facial expression game tried to show correct facial expressions to achieve more scores, we cannot guarantee all the images are correctly labeled. To reduce the impact of those weakly labeled images, we propose to use yet another ”recursive” step to cleanse the datasets.

In our data cleaning step, we use the corresponding trained models to predict the facial expression scores for all the images in the CIFE and GaMO datasets, respectively. So for each image, there are 7 facial expression scores related to it. Since we know the facial expression label of each image, only one of the 7 facial expression scores is useful for that image. For example, a Happy image will result in 7 facial expression scores describing the 7 probability of this image being classified as one of the 7 facial expression types, but we only consider the value obtained for Happy facial expression. Using the facial expression scores, we can sort all the images in each of the seven facial expression classes. For example, to sort all the Sad

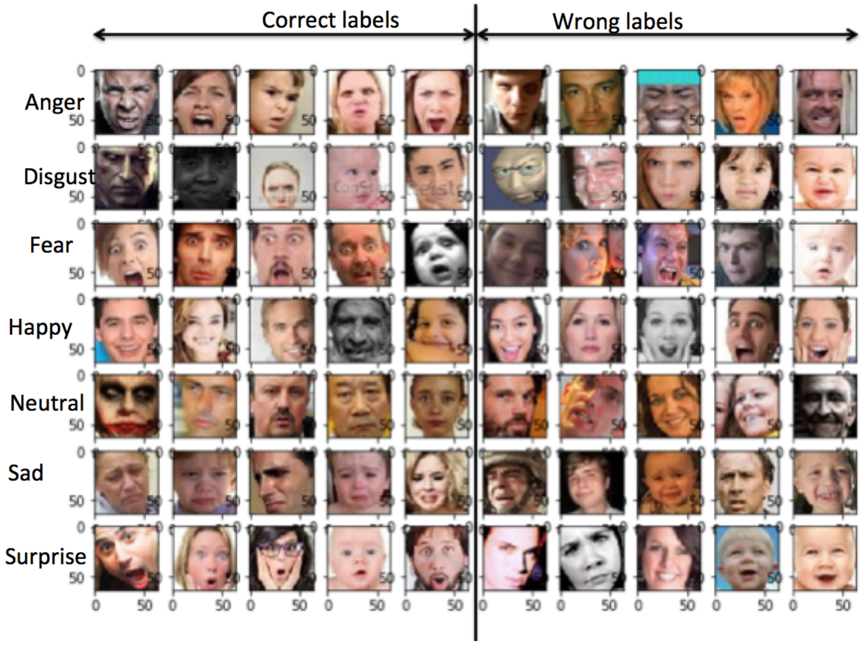


Figure 7.3: Comparison of the "well" labeled (Correct) and "badly" labeled (Wrong) samples in CIFE

images, we first find all the images labeled Sad. Then we sort them based on their predicted Sad scores. In each facial expression group, a higher score means that the score assigned to the image is more reliable. To verify if this sorting process is meaningful, we randomly picked images from the top 5% "well" labeled images and the bottom 5% "badly" labeled images. Examples of these images are shown in Figure 7.3 and 7.4, respectively. From these two figures, we can see the quality of the original labeling. The "good" labeled samples tend to have better facial expression correspondence with their labels, while for the "badly" labeled samples, the image content are less related to their labels.

Since the facial expression scores show how correctly images are labeled, we can use the predicted scores as criteria to "clean" each dataset. Here we excluded the 10% facial expression images with the lower facial expression scores in each facial expression class in both CIFE and GaMo. That yields in the "cleaned" CIFE and GaMo datasets. To see if this "self-cleansing" can improve the facial expression recognition performance, we finetuned the pre-trained facial expression models using the updated CIFE and GaMo dataset. After about 5000 iterations of training with 50 images as one batch, we stopped at the converged models. The confusion matrices for CIFE and GaMo are illustrated in Table 7.9 and Table 7.10. By comparing them to the corresponding confusion matrices in Table 7.3 and Table 7.4, we can see the average facial expression recognition accuracy values for CIFE and GaMo increase by 9% and

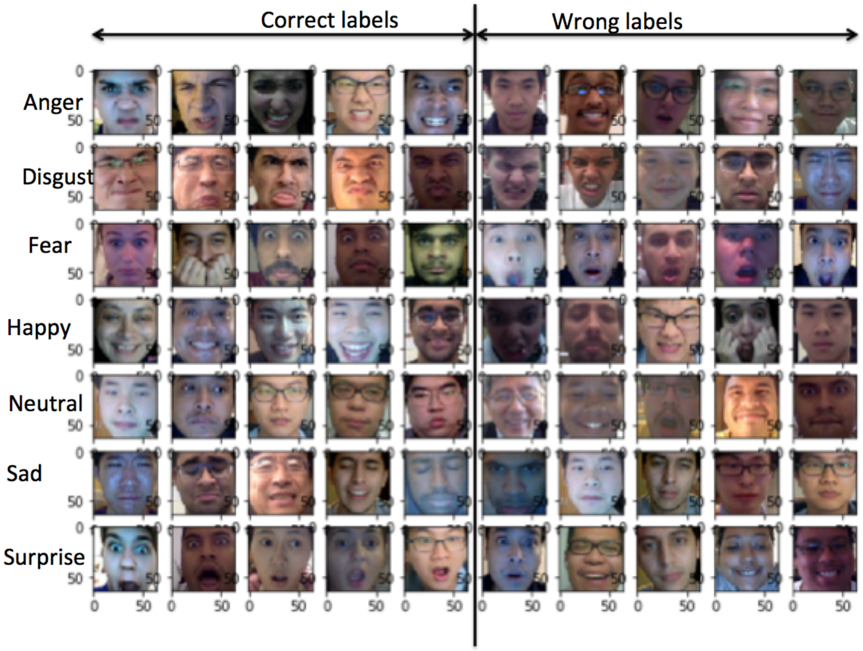


Figure 7.4: Comparison of the "well" labeled (Correct) and "badly" labeled (Wrong) samples in GaMo

8% respectively, from 76% and 75% to 85% and 83%. We can draw the conclusion that by self-cleansing the facial expression datasets, we obtain cleaner datasets that consist of facial expression images with higher quality, which can contribute to more accurate facial expression recognition.

The changes in different facial expression classes can be seen in 7.9 and Table 7.10 after the data cleansing. Comparing the confusion matrices of CIFE and GaMo recognition results before and after data cleansing, we find most of the expressions have high recognition rates. Note that for disgust expression of the updated CIFE model, the recognition rate dropped to 29% (from 53% in Table 7.3) . This is most probably due to the data imbalance of the smaller training dataset after the data cleansing, since number of the samples for disgust is even smaller, lower than what the CNN model expected to have, in a training process to control the overal loss of all the expression categories. Nevertheless, the recongition rates with the model cleaned GaMo data have consistent improvement across all facial expression categories (except a slightly decrease in Neutral), since the GaMo data is more balanced. This is one more reason why we have collected the GaMo dataset.

Table 7.9: Self evaluation confusion matrix of the cleaned CIFE

	Ang	Dis	Fea	Hap	Neu	Sad	Sur
Ang	0.91	0.01	0.02	0.01	0.03	0.02	0.01
Dis	0.1	0.29	0.04	0.07	0.30	0.11	0.06
Fea	0.03	0.01	0.78	0.02	0.04	0.02	0.08
Hap	0.01	0.01	0.0	0.99	0.01	0.01	0.0
Neu	0.02	0.08	0.08	0.05	0.85	0.02	0.03
Sad	0.02	0.01	0.01	0.003	0.004	0.94	0.0
Sur	0.01	0.01	0.08	0.02	0.07	0.01	0.85

Table 7.10: Self evaluation confusion matrix of cleaned GaMo

	Ang	Dis	Fea	Hap	Neu	Sad	Sur
Ang	0.76	0.06	0.02	0.03	0.06	0.04	0.02
Dis	0.06	0.76	0.02	0.03	0.05	0.06	0.01
Fea	0.01	0.03	0.73	0.02	0.05	0.01	0.12
Hap	0.01	0.01	0.01	0.90	0.04	0.01	0.01
Neu	0.01	0.02	0.03	0.04	0.82	0.04	0.02
Sad	0.02	0.05	0.01	0.05	0.01	0.78	0.01
Sur	0.01	0.01	0.05	0.01	0.007	0.002	0.91

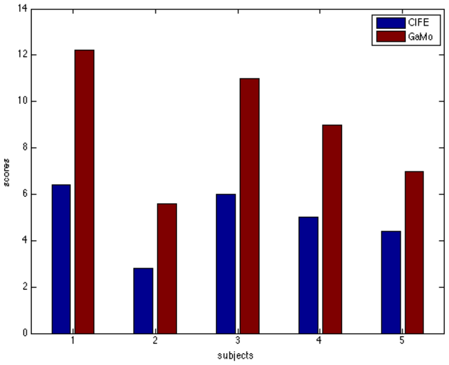


Figure 7.5: Users' average scores on two GaMo and CIFE based CNN models

7.5 Comparison in user feedback

The goal of facial expression recognition research is often to train a model that can perform well in real scenes. This is especially true in human-computer interaction applications for real daily activities, such as satisfaction studies of customers and viewers, and assistive social interaction for people in need, or individuals with visual impairment and autism spectrum disorders (ASD). One approach to verify an expression detector is through a test on ordinary people with natural facial expressions. To accurately evaluate the two models, we analyze the data collected from five new users (3 male and 2 female) who are not included in the GaMo dataset, while playing the general version of the game. Note that in the phase of GaMo data collection, we mainly use the customized game interface since users cannot perform well with the general game interface. In this game engine performance study, the general game is played five times by each user with the same game settings and the scores of the five rounds are recorded. Using the two versions of our game engine, one trained on CIFE and the other on GaMo, respectively. Figure 7.5 shows the result of this experiment. We have plotted the two average scores for each player on games powered by the two game engines. According to this figure, the GaMo game engine has a much better performance and results in higher scores. This further confirms that the model trained on GaMo is more suitable for real-world expression recognition.

This result agrees with the cross testing results which show that the GaMo model has a better performance on the GaMo dataset itself. These observations would also support our claim that GaMo is very useful in detecting subtle expressions. For instance, the user can gain a point with a normal smile expression in GaMo model game as shown in Figure 7.2, while using the CIFE model, the expression can not be detected. Same fact holds for detecting anger or any other expressions, as our players do not have any prior knowledge of how obvious and explicit their facial expression should look like.

8 Facial Expression Recognition: an Application

We have shown that our proposed recursive framework is useful in obtaining facial expression data and training a robust expression predictor. We also believe that the data and trained model is useful when solving real problems. So in this chapter we present the results of a facial expression application (App) to help people with Autism Spectrum Disorder (ASD) learn better social interaction skills.

8.1 A facial behavior training game for people with ASD

Autism Spectrum Disorder (ASD) is a group of developmental disabilities that are characterized by social-communication impairments that cause significant deficits in the areas of social interaction, communication and language skills, and repetitive behaviors and interests. In particular, these deficits cause difficulties in the perception of faces and the expressions of faces, understanding facial expressional states, and the perception of gaze direction. Studies of the face processing skills of people with ASD have shown that these impairments are widespread and present from an early age, and affect both the perception and memory of faces [67]. Studies have demonstrated that computer-based instruction is more effective in individuals with ASD versus traditional instruction [68, 69, 70, 71]. Due to recent advances in computer vision and deep learning, new avenues are now being explored in computerized assistive and intervention programs for people with disabilities.

Our deep learning model can have high accuracy in analysis people's facial expressions, so we want use the facial expression deep model to help the ASD group. The main problem for the ASD people is correctly recognize and perform facial expression, so our task is to train them notice the facial expression and also learn to express right facial expressions. Based on the target we want attain, we name our project EmoTrain.

8.2 Related work in applications for ASD

There exists a lot of interesting work relating to applications of human facial expression recognition. The MIT Media Lab has created "Affectiva", an application for analyzing your smile, which began as an effort to help people on the autism spectrum who have difficulty reading facial expression, and is now being commercialized to help businesses understand their customers[68]. In [69] Cockburn, et al propose including real-time expression recognition in an existing dynamic game for children with ASD as a means for improving the efficacy of the gamefied intervention platform. In their work they utilize the Computer Expression Recognition Toolbox (CERT), which analyzes facial expressions in real-time and can classify them into 7 basic facial expressions and 30 facial action units from the Facial Action Coding

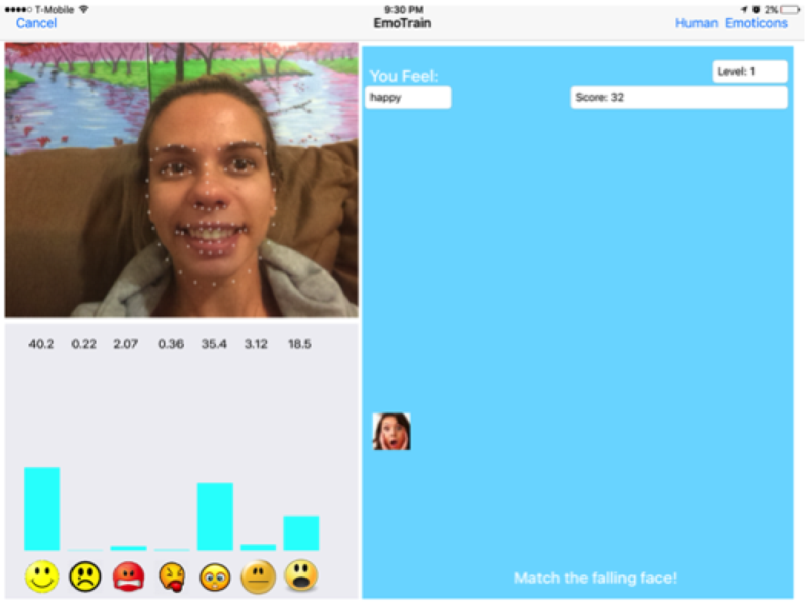
System. In [70] Tanaka, et al propose Let's Face It!, a computer-based intervention program, that is comprised of 7 interactive computer games. These games are created to enhance the face recognition skills, holistic face processing skills, and attention to the eye region in children with autism. The results from their study were promising, indicating that a short-term intervention program produces significant improvements in the face processing skills of children with autism. In [71] Golan, et al evaluate an animated series designed to enhance facial expression comprehension in children with autism spectrum conditions, The Transporters. The children who participated in this study exhibited significant improvement in facial expression recognition in all the task levels.

Studies have demonstrated that there are several advantages to using computer software for instruction with ASD individuals. In [73] the impact of computer instruction vs traditional behavioral techniques for vocabulary acquisition for children with ASD was examined. The study demonstrated that the children learned more, paid more attention, and were more motivated in the educational software program that was designed based on behavioral learning principles than in an education program with human instructors. In [74] a computerized intervention program for teaching adults with ASD to recognize facial expressions in faces and voices, Mind Reading, was evaluated. The adults who used Mind Reading for even a relatively short period of time showed significant improvement in their facial expression recognition skills. The use of educational software for individuals with ASD is more effective for a myriad of reasons. Individuals with ASD prefer computerized educational environments since they are predictable and consistent, and devoid of the social factor which causes them stress. With computer-based instruction these individuals have the ability to work at a pace that suits their learning capacities, and can repeat lessons until they are mastered [72].

8.3 EmoTrain: system overview

EmoTrain is a platform that is designed to target deficits in recognizing and labeling facial expressions, and reciprocating facial expressionally and to help teach these face processing skills to individuals with ASD. EmoTrain is designed to teach individuals with ASD to recognize 7 basic human facial expressions and to also perform these 7 facial expressions. We use the interaction pattern in our facial expression games. When playing EmoTrain, users are asked to attempt to match the images of face expressions they are presented with on the screen. By incorporating our work in real-time facial expression recognition from a mobile camera, the platform is able to track the user's expressions in real-time and judge if the user is performing those expressions correctly. In [69] it is demonstrated that training in facial expression mirroring is essential to developing recognition skills. This sort of training in mimicking expressions allows us to improve the skills of individuals with ASD in facial expression recognition, and in automatic facial expressional reciprocity.

The game logic for EmoTrain is as follows: a user is presented with images of various faces representing the seven different facial expressions and has to perform that same expression in order to score. The EmoTrain interface is shown in Figure 8.1. On the right half of the screen is the game scene, on the top left corner is the live video stream from the front-facing camera of the device, and on the bottom left corner is displayed a visualization of the facial expressions detected in the user?s face in real-time.



You Feel: happy

Level: 1

Score: 32

Match the falling face!

Figure 8.1: Screenshot of EmoTrain interface.

Each face image target enters the screen from the top and the user has to match that facial expression before it reaches the bottom in order to score. The face image disappears if the user performs a facial expression that matches the target image, which is judged by the CNN-based facial expression detector. The EmoTrain platform accesses the front facing camera on the user's device and sends the camera frames to our server running the CNN model– which analyzes each image and generates a probability vector for the seven facial expressions and sends this back to the app. The app then compares the probability vector with the image label for the face image target. In order to combat inaccurate facial expression probability results and make it easier for a user to score each facial expression category is assigned a predefined threshold. This threshold varies for each facial expression since some facial expressions are more difficult to mimic voluntarily such as anger and fear. If the threshold is reached for the facial expression category that is the same as the image target label then that is considered a match and the user scores.

In order to make the platform more engaging and interactive for the user we have added certain visualizations and multimedia. On the real-time video of the user's face on the upper left corner. Dlib's face landmark detector [78] are utilized to track the user's face on the screen and 68 landmarks on the face. On the lower left corner of the screen a bar graph live visualization is displayed that shows the probability of each facial expression the CNN model predicts for the user's face.

8.4 User evaluation

In order to gauge the effectiveness of the EmoTrain platform, 9 subjects participated in an evaluation study. The participants were all young adults (ages 18-25) who were diagnosed with ASD. The participants received training with the EmoTrain platform 2 times a week for 2 weeks, for 20 minutes each session. In total each participant received 80 minutes of training in the four 20-minute sessions. The participants were administered a pre-use survey before they began using the platform and a post-use survey after they had completed all their four training sessions. Both surveys consisted of two sections: the first section was a general survey to gauge interest and the second section was an assessment to determine the subject's facial expression recognition abilities. The general part of the pre-use survey administered before a subject started using EmoTrain was about the subject's online gaming preferences, and the types of mobile devices they use along with their proficiency in using them. The assessment part of the pre-use survey asked a user to identify the facial expressions in 14 images, where there were 2 images for each of the 7 basic facial expression categories. The assessment questions of the pre-use surveys were simple: How is he/she feeling? The assessment was multiple choice with 7 choices for each facial expression and an 8th choice of "I'm not sure". The general part of the post-use survey administered after a subject completed all his/her sessions with EmoTrain was about the subject's experience with EmoTrain such as difficulties with the platform and recommendations for improvement. The assessment part of the post-use survey (the post-assesment) asked a user to perform the same expression identification task as the pre-assessment but with a different set of 14 images.

The effectiveness of the platform was analyzed by determining if the facial expression recognition skills and the facial reciprocity skills of the participants improved. The improvement of the facial recognition skills of the subjects was determined by looking at the subjects' responses to the assessments administered before and after the study. The improvement of the facial reciprocity skills of the subjects was determined by looking at the subject's score for each session. If the score increased significantly for each session with EmoTrain then we concluded that the facial reciprocity skills for that participant had improved. In Figure 8.2

we compare each subject's session 1 total score with session 4 total score. From this we see that all the participants' total score increased each session of game play with EmoTrain. This demonstrates that even with a relatively short training period of 80 minutes the facial expression recognition and expression reciprocity skills of the participants in this study significantly improved. In Figure 8.3 we see each subject's pre-assesment in comparison to each subject's post-assessment survey. Every subject was scored out of 14, which is the total images asked to identify. From this figure we see that most of the participants' scores in the post-assessment show a significant improvement from their scores from the pre-assessment in facial expression recognition.

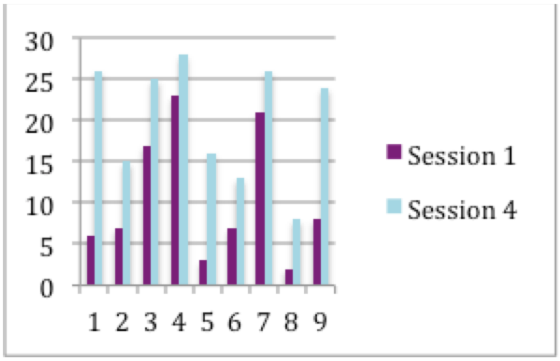


Figure 8.2: The improvement of subjects' facial reciprocity skills: comparison of their game scores in the first session and the last session.

The general survey that was administered to the participants upon completion with the sessions demonstrated that 100% claimed to really enjoy the game, and 90% would recommend EmoTrain to a friend. Most of the participants found the interface fairly easy to understand and use, with only 20% stating that it was difficult to match the face expression targets.

8.5 Conclusion and discussion

As a conclusion, EmoTrain is designed as an interactive platform based on deep learning that is designed to help improve face expression recognition and reciprocity skills in individuals with Autism Spectrum Disorder. EmoTrain is a gamefied training platform that prompts user's to match the face expression images shown with their own faces in real-time. By utilizing our work in real-time facial expression recognition from a mobile camera, EmoTrain's interface tracks the user's face from the device's front facing camera and can determine if the user makes the correct face expression. Using this method of mimicry training, the objective is to help people with ASD overcome their impairment in face processing skills such as perception of faces, understanding facial expressional states, and facial expressional

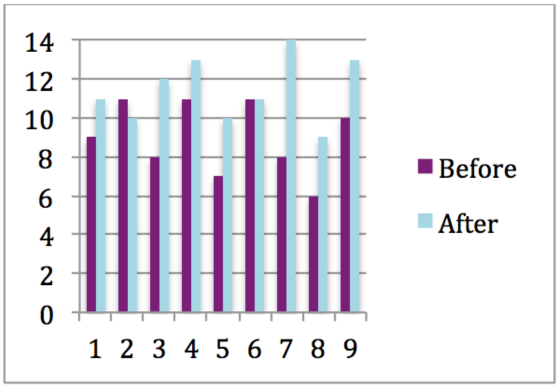


Figure 8.3: The improvement of the facial recognition skills of the subjects: comparison of pre-assessment and post-assesment.

reciprocity. The effectiveness of this platform was evaluated by administering training with EmoTrain to a group of participants diagnosed with ASD. By analyzing the data collected from the participants' performance while playing on EmoTrain over time and comparing their face processing skills before and after playing, we demonstrated that the platform is effective.

Part II

Action Unit Detection: Learning Transfer, Attention Coding and Temporal Fusion

9 AU Detection: An Overview

Facial Action Unit (AU) detection is an essential step in the facial analysis. With a robust AU detector, facial expression and facial related action problems can be solved more effectively. AU detection is the process to find some basic facial actions defined by FACS, the Facial Action Coding System [26]. Each AU represents a basic facial movement or expression change. Table 4.1 listed 12 basic AUs labeled in the BP4D dataset [52], and Figure 9.1 shows four basic AUs, namely eyebrows lower, cheek raiser, chin raiser and lip tighter. The AUs are elements for more complicated facial actions. For instance, sadness might be the combination of AU1 (inner brow raiser), AU4 (brow lower), and AU15 (lip corner depressor).

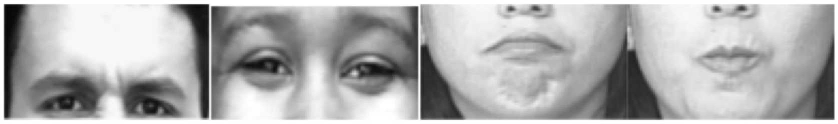


Figure 9.1: Action unit images for AU 4, 6, 7, 17

Most of current AU detection approaches either need the processed faces with frontal views or the texture features are artificially designed, making the features not well learned [37, 47]. To tackle these problems, we propose the EAC (enhancing and cropping) Net, (Figure 9.2), based on the convolutional neural network (CNN)[33] to detect facial AUs automatically. Even though a CNN has great capability in finding different patterns across images, it is not flexible enough to know which regions of the images need more attentions, and when learning from images in a dataset, the network is unable to shift the pixels to compare across corresponding regions. Therefore we enhance the regions of interest by assigning higher learning weights to corresponding areas during deep model training, and then crop corresponding areas to force the network to learn better representations through training on related individual regions. We first build the enhancing net (E-Net), which is constructed by adding attention layers to a pretrained VGG net[33], one of the very effective CNNs. The E-Net yields a significant improvement in the average F1 score and accuracy on the BP4D dataset as compared to the state of the art approaches. We then add cropping layers on top of the E-Net and design the EAC-Net. The cropping layers are implemented by cropping AU areas of interests from high-level convolutional feature maps. The EAC-Net yields up to 7.6% increase in average F1 score and 19.2% improvement in accuracy as compared to the state of the art approaches when applying to the BP4D dataset.

To better address the problem of effective fusion of temporal information in AU detection, we also propose a C-Net based region of interest (ROI) adaptation optimal LSTM-based temporal fusing approach. The optimal selection of multiple LSTM layers to form the best

Table 9.1: Rules for defining AU centers

AU index	AU Name	AU Center
1	Inner Brow Raiser	1/2 scale above inner brow
2	Outer Brow Raiser	1/3 scale above outer brow
4	Brow Lowerer	1/3 scale below brow center
6	Cheek Raiser	1 scale below eye bottom
7	Lid Tightener	Eye center
10	Upper Lip Raiser	Upper lip center
12	Lip Corner Puller	Lip corner
14	Dimpler	Lip corner
15	Lip Corner Depressor	Lip corner
17	Chin Raiser	1/2 scale below lip
23	Lip Tightener	Lip center
24	Lip Pressor	Lip center

LSTM Net is carried out to best fuse temporal features. The proposed approach outperforms the state of the art significantly, with an average improvement of around 13% on BP4D and 25% on DISFA, respectively.

This part is organized as follows. Chapter 10 provides an overall summary of the ideas of our EAC-Net approach, and some related work. Chapter 11 details the design and analysis of the EAC-Net. A temporal fusion approach based on LSTM is described in Chapter 12. To evaluate our method, we test our approaches with standard datasets in Chapter 13 and with our candid image dataset CIFE-AU in Chapter 14. Possible AU detection based applications are discussed in Chapter 15.

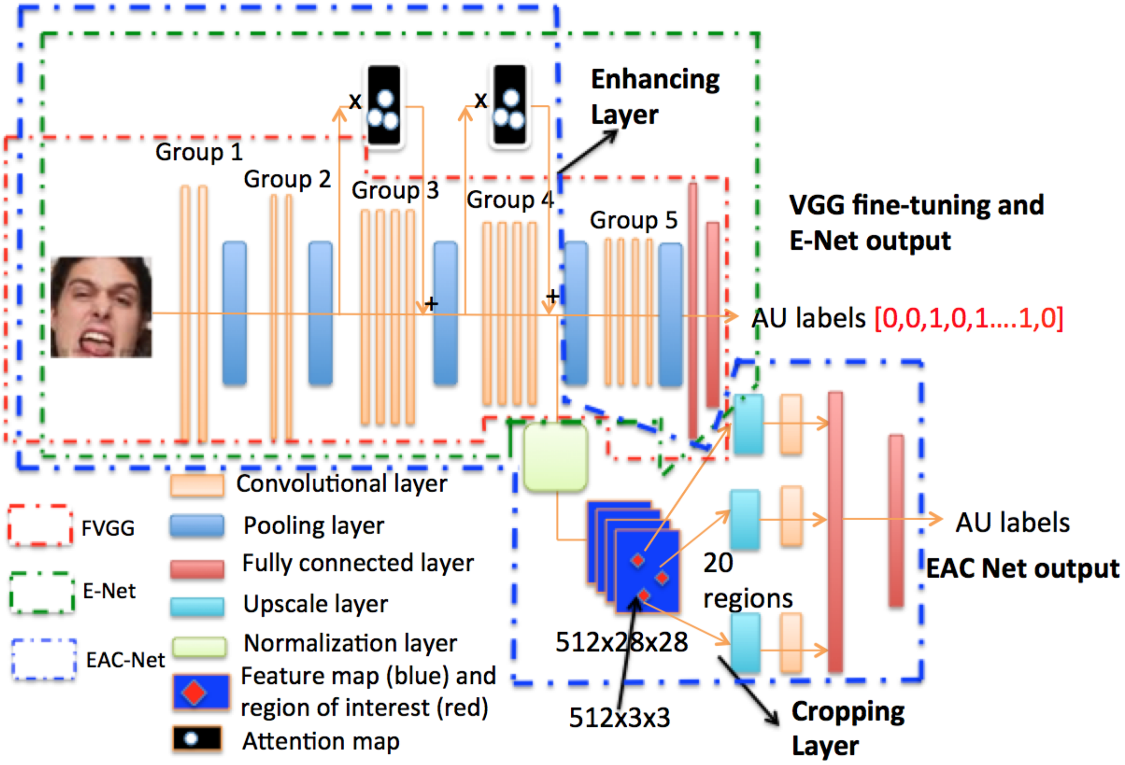


Figure 9.2: The structures of FVGG, E-Net and EAC-Net for AU detection

10 AU Detection: Design Considerations and Related Work

10.1 AU detection: design considerations

The idea of our approach is inspired by recent breakthroughs in deep learning research. The first inspiration is learning transfer. Pretrained ImageNet models and transfer learning have found significant applications in many areas [27, 28]. The main reason is that the low-level convolutional layers extract similar features across different recognition tasks, which means the learned filters can be transferred. The second inspiration is attention coding and cross-layer connections. Yang et al [29] proposed to use an attention layer for finding interesting areas to provide better answers in a visual question-answering task. In salient object detection [30], a salient map is used to describe the important sub-areas of an image. This salient map (or attention map) is a 2D matrix with its elements ranging from 0 to 1, depicting the importance of corresponding pixels on the entire image. We believe that by applying an attention map to an existing network for facial computing, richer and more robust information can be obtained. A skipping layer structure is used in the Residual Net [51]. Since both lower and higher levels of CNN layers correspond to more spatial and semantic information, the skipping connection of layers feeds both spatial and semantic features into CNNs, which can improve the model performance. The third inspiration is the region of interest (ROI) learning. The SPP Net [31] proposed an ROI pooling idea, which can turn different areas of interest into fixed length features. Faster RCNN [32] generated region proposals for objects to be detected and used the proposals to determine the “objectiveness”. These findings made us believe that it might be possible to apply a similar operation to facial interest regions and learn individual filters for specific regions.

The design of our EAC-Net has integrated the three ideas: learning transfer, attention coding and cross-layer connections, and ROI learning. Figure 9.2 shows the structure of our EAC-Net. The EAC-Net consists of three key parts: a pretrained model, enhancing layers, and cropping layers. The first part is a fine-tuned pretrained VGG 19-layer network [33]. The low-level convolutional layers (Group 1 and 2) of the pretrained network and their parameters are frozen for low-level visual feature extraction. The parameters of the higher-level convolutional layers (Group 3 and 4) are updated for the AU detection task. The use and fine-tuning of VGG pretrained net are with the spirits of learning transfer, to ensure that the network has a deep understanding of the input images. The second part of the EAC-Net consists of the enhancing layers that are on top of Group 3 and 4 convolutional layers, thus integrating attention coding and cross-layer connecting. The purpose of adding these layers is to give more attention to individual AU areas of interest, meanwhile fusing features cross different layers. The features extracted after the enhancing layers are supposed to contain more valuable information by integrating lower level spatial features and higher level semantic features for AU detection. The third part of the framework includes the cropping layers for regions of interest learning. Sub-features are cropped from ten selected interest areas of the feature map, followed by upscale layers and individual convolutional layers for each sub-area for further regions of interest learning. The purpose of the cropping layers is to assure that only corresponding regions are compared by the individual cropping networks. Adding cropping layers as the higher convolutional layers would also help the region to obtain deeper contextual information.

10.2 AU detection: related work

AU detection has been studied for decades and many approaches have been proposed. Facial key points play an important role in AU detection. Many conventional approaches [35, 36, 39, 41, 42, 43, 49, 55] were designed by employing texture features near the facial key points. Valstar et al [34] analyzed Gabor wavelet features near 20 facial landmark points. The features were then selected and classified by Adaboost and SVM classifiers. Since landmark-based geometry changing is robust in many AU detection methods, Fabian et al [37] proposed an approach for fusing the geometry and local texture information. The geometry information is obtained by measuring the normalized facial landmark distances and the angles of the Delaunay mask formed by the landmark points. The texture features were obtained by applying multiple orientation Gabor filters to the original images. Zhao et al [38] proposed the Joint Patch and Multi-label Learning (JPML) for AU detection. Similarly, landmark-based regions were selected and SIFT features were used to represent the local patch. Overall, the conventional approaches focused on designing more representative features and fine-tuning more robust classifiers. In addition to facial AU detection, some studies also have focused on other facial related problems. Song et al [48] investigated the sparsity and co-occurrence of action units. Wu [44] exploited the joint of action unit detection and facial landmark localization and showed that the constraints can improve both AU and landmark detection. Girard et al [45] analyzed the influence of different sizes of training datasets on appearance and shape-based AU detection. Gehrig et al [46] tried to estimate action unit intensities by employing linear partial least squares to regress intensities in AU related regions.

Over the last few years, we have witnessed that CNNs boost the performance in many computer vision tasks. Compared to most conventional artificially designed features, CNNs can learn and reveal deeper information from the training images which in turn contribute to better performance. Zhao et al [47] proposed a region CNN-based approach for AU detection. Instead of directly applying a regular CNN to the entire input image, the network divides the input image into 8x8 blocks and then trains over these regional areas independently. All sub-regions are then merged back into one net, followed by regular convolutional and fully connected layers. The proposed approach outperforms both the regular and Fully Convolutional Net (FCN)[47]. Jaiswal et al [40] proposed using a CNN with a shallow region and shape mask a is employed to learn static CNN features while Long and Short Term Memory (LSTM) is used to extract dynamic features from the trained CNN mode.

Our proposed approach is also CNN-based and we share similar ideas with Zhao et al [47] in training sub-regions independently. The distinction of our approach is that instead of directly dividing the image into blocks, we use a “smarter” way to find the important areas and crop the regions of interest for individual training. In their approach, the facial landmarks play an important role for normalizing the faces. Errors will accumulate in both landmark detection and face normalization processes, and facial normalization may neutralize expressions. However, in our approach, the network directly works on the interest areas of original face images, and even though landmarks are also used for building the attention map, our approach has a large tolerance for landmark shifting since we use a relatively large local region to cover the AU target areas. This will reduce errors from misalignment along images and in the meantime focus more on interest regions.

11 EAC Net: CNN Learning Transfer and Attention Coding

The EAC-Net is composed of three parts: the fine-tuned VGG network, enhancing layers, and cropping layers. For comparison purposes, we implement three networks using VGG net as their base: FVGG, the fine-tuned VGG network; E-Net, the Enhancing Net based on the fine-tuned VGG; and finally EAC, the integration of Enhancing and Cropping Nets based on the trained E-Net model.

11.1 FVGG: fine-tuned VGG model

Fine-tuning pretrained models for image classification are proved to be efficient in many areas [27, 28]. To make sure that we can have a deep understanding of the images for AU detection, we employed the VGG 19-layer model [33]. The VGG model follows a conventional CNN structure, comprising 16 convolutional layers and 3 fully connected layers. It also includes 5 pooling layers downsampling input images from 224×224 to eventually 7×7 feature maps. In our implementation, we divide the convolutional layers into 5 groups as shown in Figure 9.2, which are separated by the 5 pooling layers. There are 2, 2, 4, 4, 4 convolutional layers from Groups 1 to Group 5, respectively. The fine-tuned VGG (FVGG) structure reflects the spirit of learning transfer as VGG has been modified in many related tasks such as object detection or facial expression recognition. Before designing our special purpose networks (E-Net and EAC-Net) for AU detection, we first modify and fine-tune the VGG net as our baseline approach. We keep the parameters of the first three groups of convolutional layers unchanged and update the rest of the layers during training. In order to match with our AU detection tasks, the numbers of nodes for the last 2 fully connected layers are changed to 2048 (by reducing parameters) and 12 (by matching the 12 AU targets), respectively. Dropout is also applied on both new layers to prevent overfitting during training.

11.2 E-Net: the enhancing net

In a regular CNN, when an image is fed into the network, CNN filters will slide through the whole image, and different image regions are processed equally. This is fine for object classification task since we have no idea of how the input image looks like. But for structured images, such as faces, we would like to give more attention to more interesting regions.

As one of our inspirations, Yang et al [29] applied an attentional layer to CNN middle features when tried to find a more effective feature for a question answering task. As shown in Figure 11.1, if we want to know what is in the basket, a simple way is direct "filter out" the basket region by applying an attention layer. However, this will discard a lot of information that is not included in the attention layer, therefore we need a method to combine both attention enhanced features and the rest of the less important areas. Another interesting approach for combining CNN multiplayer features is the skipping layer structure in the Residual Net [51]. We illustrate the skipping layer structure in Figure 11.2 (left). The skipping layer connection of the residual net makes the CNN incorporating both lower level spatial features and higher level semantic features as input, together providing a better representation of the input image. Integrating these two ideas, we designed our E-Net unit as the combination of the regions enhanced feature and the regular filtered feature as shown in 11.2 (right), which is an effective integration of the attention layer and skipping layer structures.

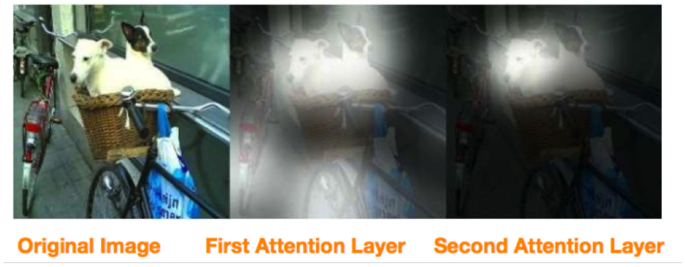


Figure 11.1: Attention layer for a question answering task [29]

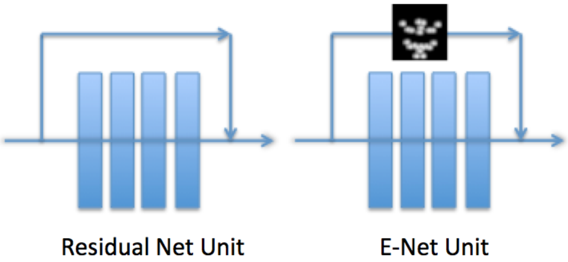


Figure 11.2: Skipping layer connection in Residual Net (left) and E-Net (right)

Figure 9.2 includes the E-Net structure and Figure 11.2 demonstrates how the E-Net works by using our enhancing layers. The feature map output from Group 2 is multiplied by the designed attention map – the first enhancing layer (details will be discussed below), in parallel with the convolutional layers in Group 3. The two feature maps – one from the enhancing layer and the other from the Group 3 convolutional layers – are then fused by element-wise summation. The same operation is performed jointly by the second enhancing layer with the convolutional layers in Group 4. The reason why we designed the enhancing layer is that not all the areas of a facial image are equally important for individual AU detection. We can see that different AUs focus on corresponding sub-areas of the face. For example, in Figure 11.3, the eyebrow raiser AU is close to the texture in the area near the middle of the eyebrows and the lip corner AUs are determined by the texture information around lip corners. These areas are more important than the nose or most of the other parts of the face. For this reason, we build the attention map for AUs based on key facial landmarks, as shown in Figure 11.3.

AU center definitions. We have noticed that many previous works provide robust facial landmark positions [50]. Furthermore, our approach does not require the localization of landmarks to be very accurate in pixels since we are trying to find the areas for AUs. We work on 12 AUs as listed in Table 9.1 since these are the ones labeled in the datasets we use. After obtaining the facial landmarks as shown in Figure 11.3, we can define the centers for AUs and then build a bounding box around the center. Observing the AU figure, we manually define the center of AUs (the green spots) based on the muscles of a human face. Note that many AU centers are not directly on the same spots of the detected landmarks. We define a scaled distance as a reference for facial pixel shifting by calculating the distance of the outer corners of the two eyes, as shown in Figure 11.3. The centers for 12 listed AUs in Table 4.1 are illustrated in Figure 11.3 (the green spots). Since most AUs are symmetric on a human face, we define one pair of points for each AU. We should note that some AUs share the same centers, such as the lip corner puller and the lip corner depressor. So finally we defined 20 AU centers on the face for the 12 AUs listed in Table 9.1. The rules for defining the centers of the 12 AUs are also summarized in Table 9.1. After obtaining the AU centers, we can build the attention map based on center positions.

AU center localization and AU area extraction. Given an image like the one shown in Figure 11.3 (left), we first obtain the landmarks for the key points on the face, which are shown with blue points. Having the facial key points, we can obtain the AU centers by shifting a distance or directly using existing facial landmarks. The AU centers are illustrated with green points in Figure 11.3. The AU centers are in pairs due to the symmetry of the human face, with each AU center corresponding to one or more AUs. To make the shifting distance more adaptable to all face images, we define a measurement reference for the shifting distance. Inner corner distance is used as the scaled-distance, as shown in Figure 11.3. This scaled-distance (listed in in Table 9.1 for each AU) is used to help locate the AU centers. We first resize the images to 100x100 to make sure the same scales are shared among all images. Then, for each AU center, we define the nearby 5 pixels belonging to the same area, therefore the size of each AU area is 11x11. Higher weight is assigned to the closer points to the AU center. The relationship follows the following equation:

$$w = 1 - 0.095 \cdot d_m \tag{11.1}$$

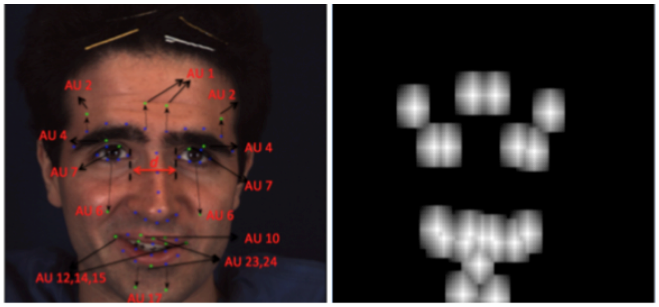


Figure 11.3: Attention map generation. Left: landmarks(blue) and AU centers (green) on a face; Right: attention map of the face

where d_m is the Manhattan distance to the AU center.

Attention map generation and application. An attention map obtained for a face image is also demonstrated in Figure 11.3. The areas in the attention map with higher values correspond to the AU active area in the face image and can enhance deep learning at these areas in our enhancing layers. We then apply the obtained attention map to the VGG feature maps. Figure 9.2 has shown the E-Net structure in the EAC-Net framework. For Group 1 and Group 2, we keep the layers unchanged for detecting low-level features. For Group 5 (size 14×14), the feature maps are too small to use any attention map, so eventually, we apply the attention map only to Group 3 and Group 4, thus we add two enhancing layers. Adding attention layers directly to the feature maps by replacing the original ones will lose all the contextual information. So, we add the attention maps to the feature maps in parallel with the convolution operations, in Group 3 and Group 4, respectively, as shown in Figure 9.2. The element-wise summation is then conducted to obtain enhanced feature maps. We call our enhancing net based on the fine-tuned VGG model the E-Net. The E-Net structure is also similar to Residual Net [51] but is designed not only for integrating features of different levels but also for generating enhanced features by applying an attention map. Of course, the E-Net also includes the VGG fine-tuning network. After training this model, we observed that the E-Net can lead to 5% increase in average F1 score and 19% increase in average accuracy on the BP4D AU dataset [52]. The detailed experimental results are reported in Chapter 13.

11.3 EAC-Net: the integrated model with enhancing and cropping

The E-Net can generate the features with more emphasis on the AU related regions, but it does not change the fact that the same AU areas are not normalized across different images, and different facial regions still share the same set of filters to extract features. For a general object detection task, sharing the same filters for the whole feature map is a norm, but for understanding a face, which is a very structured target, this is not an effective approach. Naturally different facial regions should use different sets of filters, which would be more efficient since we can train separate filters for specific AU regions, around nose, eye or mouth areas, respectively.

In the state of the art work, a simple region layer structure [47] has been proposed to provide certain local convolutional training. The idea is to divide a feature map into 8×8 subregions, and each region uses a separate CNN model. Then if faces are aligned for all the input images, we can obtain those locally trained CNN models for individual sub-blocks. The disadvantage though is that during face normalization and image division, some context information might be lost, and since face normalization is trying to align the faces to a neutral expression, the facial expression might be weakened.

C-Net construction. The goal of our cropping net (C-Net) is to obtain each individual AU related area without changing the textural information of the original images. Then, for each AU sub-area, independent convolutional layers are applied to learn more features. Figure 9.2 also includes the structure of the C-Net. Cropping layers are added to the end of Group 4, right after the enhancing feature maps are obtained. The output size for the feature map from Group 5 is 512×28×28. With the same ratio as an AU area of 11×11 pixels in the attention map versus the face image of 100×100 pixels, the cropped areas should have the size of 3×3. For each of the 20 AU centers, we obtain a feature sub-map with size 512×3×3; in total, we have 20 such feature sub-maps for the 20 AU centers. When adding convolutional layers after the cropped feature map, the newly obtained feature map size will be 512×1×1. We feel that this is less representative for the AUs. So, before adding new convolutional layers, we apply an upscaling layer to the feature maps, upscaling the feature maps to 512×6×6. Actually, our experiments show that this upscaling layer by itself leads to approximately 1% increase in AUs average F1 score.

EAC-Net construction and training. To make the C-Net converge more quickly, we build the C-Net on top of the pretrained E-net, thus leading to the Enhancing and Cropping Net (EAC-Net). So the features obtained from Group 4 have already been pretrained for AU detection. During implementation, we have found that feature values obtained from the last convolutional layer of Group 4 are very large and make the C-Net unable to converge. So a local response normalization layer is added before C-Net convolutional layers. The local response layer normalization algorithm follows Eq. 11.2:

$$x_i' = \frac{x_i}{(k + \alpha \Sigma_j x_j^2)^\beta} \tag{11.2}$$

In our experiments, k=2, α=0.002 and β=0.75. x_i, x_i' are the model extracted feature values before and after applying the normalization, while $j =, 1..., 9$ denotes the 9 neighboring 2D feature pixels around x_i (including itself). All the individual convolutional layers are followed by fully connected layers with a fixed size of 150. We then concatenate the fully connected layers. The rest is similar to FVGG and E-Net.

AU detection is different from a regular classification task in the sense that instead of classifying images into one object category, multiple AUs can co-occur simultaneously. Thus this is a multi-label binary classification problem. Cross entropy as in [47] is used to measure the loss for this kind of problem. In our loss function (Eq. 13.2), we added offsets to prevent the number from becoming too large:

$$Loss = -\Sigma(l \cdot \log(\frac{p + 0.05}{1.05}) + (1 - l) \cdot \log(\frac{1.05 - p}{1.05})) \tag{11.3}$$

where l is the ground truth label for one certain AU, p is the regressed number by the trained model for the certain AU ranging from 0 to 1.

12 LSTM Net: Temporal Fusion in AU Detection

12.1 Why LSTM

A facial action always has a temporal component when using a video sequence as the input, hence knowing the previous states of a facial expression can definitely improve the AU detection. However, one of the limitations of the CNN structure is the lack memory of previous states. Regular CNNs are only able to process a single image at a time. To deal with a sequence of images, C3D [57], which is basically a 3D version of CNN, has been proposed. C3D can deal with sequential images but the number of input images is fixed. The training of a C3D is very time consuming too. Another huge shortage is, compared to using regular CNN, the lack of existing pretrained models similar to VGG [33], GoogleLeNet [58] and ResNet [51], which can all provide very good initial parameters as a starting point for training. The current best network for temporal fusion is the Long Short Term Memory (LSTM) network [60]. As a recurrent net, it can memorize the previous features and states, which can help current feature learning and estimation. It also has novel gate structures to make it suitable for long time and short time temporal feature learning. LSTM has also proved to be effective in action recognition [59].

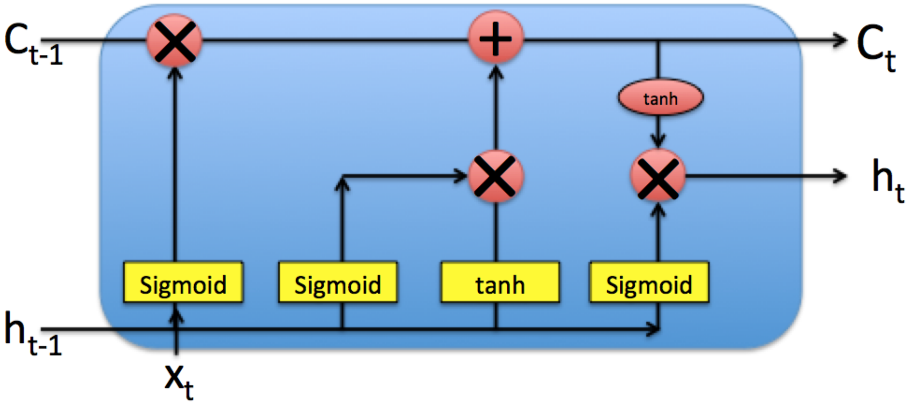


Figure 12.1: Structure of a simple LSTM block [61].

12.2 LSTM-Net: temporal fusion based on LSTM

The structure of a LSTM block is shown in Figure 12.1. In the LSTM block, C_{t-1} and C_t are the cell state parameters at the previous and the current times, the long and short memories are described by the cell state vector C_t. The cell states store the memory parameters in LSTM. At each time step, a LSTM kernel will take the previous output h_{t-1} and the new input x_t to generate the new output h_t through gates, which is shown as yellows blocks in figure 12.1. Meanwhile, the cell state gets updated. A new input feature fed to a LSTM block will go through three steps. First, the LSTM has to decide what information to obtain/forget from the old cell state. This is based on the previous LSTM output h_{t-1} and new input feature x_t. The forget vector f_t follows equation 12.1:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{12.1}$$

where W_f and b_f are the forget gate parameters. The next step is to update the cell state for future use. The new cell state C_t is determined by two elements: previous partially saved cell state C_{t-1}, current LSTM input x_t and previous output h_{t-1}. The last two vectors need to go through an “input gate” and a *tanh* activation function. The updated cell state can be obtained using equation 12.2:

$$C_t = f_t * C_{t-1} + i_t * \check{C}_t \tag{12.2}$$

where i_t is the merged input of x_t and h_{t-1} defined by equation 12.3,

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{12.3}$$

where W_i and b_i are the input gate parameters. $\check{C}_t$ in equation 12.2 is the candidate cell state for generating final cell state and output which we can regard as a temporal cell state parameter, following equation 12.4:

$$\check{C}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{12.4}$$

where W_c and b_c are the candidate gate parameter.

Finally, we generate the current output h_t for the LSTM based on the updated cell state C_t, the current input feature x_t and the previous output h_{t-1}, which can be described by equation 12.5:

$$h_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \cdot tanh(C_t) \tag{12.5}$$

where W_o and b_o are the output gate parameters. Meanwhile, the output h_t and the cell C_t are passed to next time output generation.

LSTM can be easily connected with CNNs. Fully connected layers of a CNN can be directly fed into the input of LSTM blocks. To better represent the fully connected features, multiple LSTM kernels can act as a layer to represent temporal features. As shown in Figure

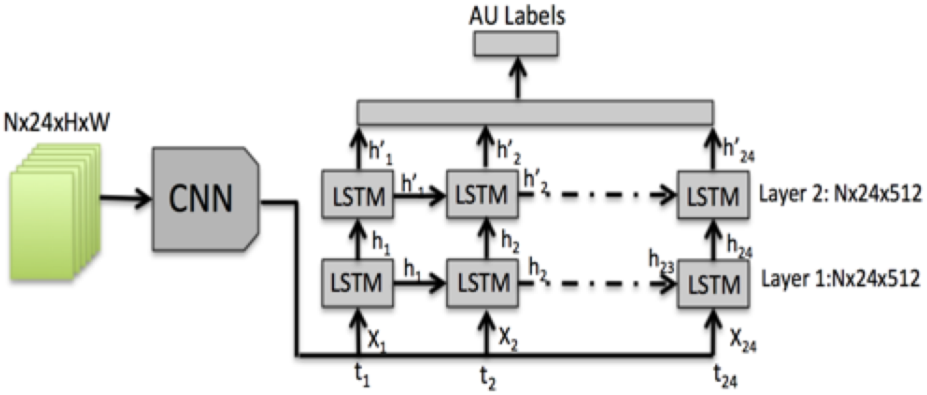


Figure 12.2: Connection of CNN and LSTM

12.2, the CNN can extract the image features as a 1-D vector. The first frame of an image sequence at time t_1 is sent to the LSTM layer at t_1. The LSTM layer will produce output feature h_1 for the first frame, then at time t_2, a new frame is sent to the LSTM layer and the new output feature is produced based on x_2 and h_1, so on so forth. Here we use $h_i(i = 1...n)$ to represent the ith LSTM feature; in Figure 12.2 $n = 24$. In different tasks, either only the last LSTM feature h_n or the whole LSTM features $\{h_1, h_2, ...h_n\}$ are used for final prediction. In our case, we believe that all the frames can contribute to the AU detection. Therefore, we use all the LSTM features; in our experiments, the number of frames is 24.

LSTM can effectively fuse the temporal information in a sequence. Similar to the convolutional layers, more than one LSTM layers can be stacked to form a LSTM-Net in order to achieve deeper understanding of the temporal relationships. As shown in Figure 12.2, the LSTM Net has 2 LSTM layers stacked for AU detection. In our work, we will try different number of layers to compare the LSTM based AU feature fusion performance.

13 AU Detection Evaluations

13.1 Datasets and evaluation methods

The most popular datasets for AU detection are CK+ [53], BP4D[52] and DISFA[54]. AU datasets are harder to obtain compared to other tasks such as image classification. This is because there are multiple AUs in one face which requires much more manual labeling work. Here we give a brief review of the AU datasets referred and compared in this thesis.

DISFA: 26 people are involved in the DISFA dataset. The subjects are asked to watch videos while spontaneous facial expressions of the subjects are obtained. The AUs are labeled with intensities from 0 to 5. We can obtain more than 100,000 AU-labeled images from the video, but there are much more inactive images than the active ones. The diversity of people also makes it hard to train a robust model.

BP4D: There are 23 female and 18 male young adults involved in the BP4D dataset. Both 2D and 3D videos are captured while the subjects show different facial expressions. Each subject participates in 8 sessions of experiments, so there are 328 videos captured in total. AUs are labeled by watching the videos, and the valid AU frames in each video vary from several hundred to thousands. There are around 140,000 images with AU labels that we can use.

The AU labels in BP4D datasets are either 0s or 1s, meaning the related AUs are active or inactive. In reality, AUs may be in states between 0 and 1, i.e., expressions have intensities. This can make the prediction more accurate but much more data labeling work will be needed. In this thesis, we mainly focus on the 0-or-1 AU label classification.

To train a deep learning model, we need larger numbers of image samples, and the diversity of the samples is also important. Similar to the experiment settings in [47], we choose BP4D to train our model. We first split the dataset to 3 folds based on subjects. Each time two folds are used for training and the third fold for testing. For the DISFA dataset, all samples are used for testing the BP4D trained model.

The balance of data is very important in training deep learning models. For our task of multi-label learning, this is even more challenging since several AUs are not independent of each other. The original occurrence rate for the 12 selected AUs is shown in the first row of Table 13.1. We can clearly see that the AUs are divided into two groups. AUs 6, 7, 10, 12, 14 and 17 are more representative than the minor AUs 1, 2, 4, 9, 11 and 12. The minor AUs are not presented in many of the image samples. If we just pick all images that do include

Table 13.1: BP4D samples balancing for AU occurrences in training

AU	1	2	4	6	7	10	12	14	15	17	23	24
Original	0.24	0.18	0.23	0.44	0.52	0.58	0.57	0.43	0.15	0.36	0.19	0.16
Minor AU In	0.56	0.43	0.40	0.47	0.57	0.64	0.59	0.56	0.35	0.58	0.46	0.39
After balancing	0.39	0.32	0.33	0.45	0.54	0.60	0.56	0.49	0.30	0.50	0.33	0.30

the less representative AUs, the occurrence rate is shown in table 13.1, the second row.

We can see that even with only the image samples that include the less representative AUs, the occurrence rates are still imbalanced. And we still need to keep the other samples to maintain the data diversity. Thus, we have finally decided to try to keep the balance of training data by changing the selection rate during training. For all the training samples, we used to equally and randomly pick a fixed number of images. To compensate for the less occurred AUs, we increase their rates during random picking operation by 4 to 7 times. Then the occurred rates by doing this are shown in Table 13.1, the third row, which is more balanced.

Both the F1 score and the average accuracy are used to measure the performance of AU detection. In a binary classification scenario, especially when samples are not balanced, the F1 score can better describe the performance of an algorithm [34, 35]. The F1 score includes two components: precision (p) and recall (r). The precision is also called the positive predictive value and is the fraction of true positive predictions to all positive predictions. The recall is also called sensitivity and is the fraction of true positive predictions to all ground-truth positive samples. Knowing p and r, we can obtain F1 using Eq. 13.1:

$$F1 = \frac{2p \cdot r}{p + r} \tag{13.1}$$

13.2 Implementation details and results

In our deep learning models, the basic VGG structure is employed as the base, so all of the input images to the networks are resized to 224x224 to become compatible with the structure. In order to maintain consistence in creating the attention map for the E-Net and the ROI cropping parameters for the cropping layers, we use the facial landmark data provided by the BP4D and DISFA datasets. We use a fixed learning rate of 0.0001 and a momentum of 0.9 in our training. For updating the model, SGD is employed. First, the VGG fine-tuned net (FVGG) is trained as the baseline for our proposed E-Net. EAC-Net training is based on pretrained E-Net with new C-Net layers. Throughout the training, 50 images are randomly

Table 13.2: F1 score comparison of AU detection on BP4D dataset

AU	LSVM	JPML[38]	DRML[47]	FVGG	E-Net	EAC
1	23.2	32.6	36.4	27.8	37.6	**39.0**
2	22.8	25.6	**41.8**	27.6	32.1	35.2
4	23.1	37.4	43.0	18.3	44.2	**48.6**
6	27.2	42.3	55.0	69.7	75.6	**76.1**
7	47.1	50.5	67.0	69.1	**74.5**	72.9
10	77.2	72.2	66.3	78.1	80.8	**81.9**
12	63.7	74.1	65.8	63.2	85.1	**86.2**
14	64.3	65.7	54.1	36.4	56.8	**58.8**
15	18.4	**38.1**	33.2	26.1	31.6	37.5
17	33.0	40.0	48.0	50.7	55.6	**59.1**
23	19.4	30.4	31.7	22.8	21.9	**35.9**
24	20.7	42.3	30.0	35.9	29.1	**35.8**
Avg	35.3	45.9	48.3	43.8	52.1	**55.9**

selected from the whole training dataset as a batch, and an epoch has 20 batches. The FVGG and E-Net models converge after about 100 epochs and the EAC-Net converges after about 500 epochs. We trained our models using a workstation with a GeForce GTX TITAN X graphics card, and it took 2-3 hours (100-150 epochs) for FVGG or E-Net and 10-15 hours (for 500-800 epochs) for EAC-Net.

13.2.1 Results on BP4D dataset

We trained three models using the BP4D dataset: fine-tuned VGG (FVGG), E-Net, and EAC-Net. The accuracy and F1 score for all 12 selected AUs, and the average accuracy and the average F1 score are listed in Tables 13.2 and 13.3, respectively. We also list the results from the state of the art work DRML [47] using deep learning and traditional approaches LSVM [47], and JPML [38] in same settings for comparison.

As shown in Tables 13.2 and 13.3, for the BP4D dataset, compared to the state of the art approaches, the VGG fine-tuned model (FVGG) has a higher average accuracy, but the average F1 score does not outperform the state of the art. Note that in our proposed approach, we do not perform any preprocessing to the input images. For the more representative AUs – 6, 7, 12, 14 and 17, the network is able to better predict the AU labels than the state of the art approaches, without being told the position of the AUs. We believe that this is due to the depth of the VGG model, which can learn very deep features, and the pooling layers, which

Table 13.3: Accuracy comparison of AU detection on BP4D dataset

AU	LSVM	JPML[38]	DRML[47]	FVGG	E-Net	EAC
1	20.7	40.7	55.7	27.2	**71.1**	68.9
2	17.7	42.1	54.5	56.0	72.9	**73.9**
4	22.9	46.2	58.8	80.5	77.4	**78.1**
6	20.3	40.0	56.6	72.3	76.9	**78.5**
7	44.8	50.0	61.0	64.1	**70.7**	69.0
10	73.4	75.2	53.6	72.4	75.7	**77.6**
12	55.3	60.5	60.8	69.1	82.8	**84.6**
14	46.8	53.6	57.0	52.8	56.7	**60.6**
15	18.3	50.1	56.2	67.4	77.6	**78.1**
17	36.4	42.5	50.0	61.2	69.3	**70.6**
23	19.2	51.9	53.9	72.2	80.2	**81.0**
24	11.7	53.2	53.9	77.0	82.3	**82.4**
Avg	32.2	50.5	56.0	64.4	74.5	**75.2**

make the AU detection robust to position and orientation shifts. For the less representative AUs – 1, 2, 4, 15, 23 and 24, however, the texture is less discriminative for instance around the eyebrow area. Also, the occurrence rates are still smaller than the other AUs even after the balancing, making the training more challenging.

The E-Net results show both better average accuracy and F1 scores than the state of the art approaches and the VGG fine-tuning net. On average, the improvement using E-Net over FVGG in the average F1 score and average accuracy are 8.3% and 10.1%, respectively. The results show the effectiveness of our proposed enhancing layers.

To explore more details of the E-Net, we extract the feature maps from multiple layers in the E-Net and VGG fine-tuning Net. In Figure 13.1, the last feature map of Group 4 convolutional layers from each of the two structures, FVGG and E-Net, are visualized. Each feature map is 512×28×28, *i.e.*—with 512 feature arrays, each of 28×28. We visualized the 512 feature map into a 32×16 arrays of 28×28 images. We can clearly see that the attention map made a big difference in the output feature maps. In the VGG feature map, the hot areas or the attention areas do not have a meaningful focus. In some areas, even the edges are highlighted as valuable features. The neural network has no idea which region to look into. While in the E-Net feature map, we can clearly see the network is concentrating on the area on the face, mainly the regions enhanced by the attention map. This can make the E-Net extract more valuable features for AU detection.

Finally, our EAC-Net achieves the best average performance in AU detection in terms of

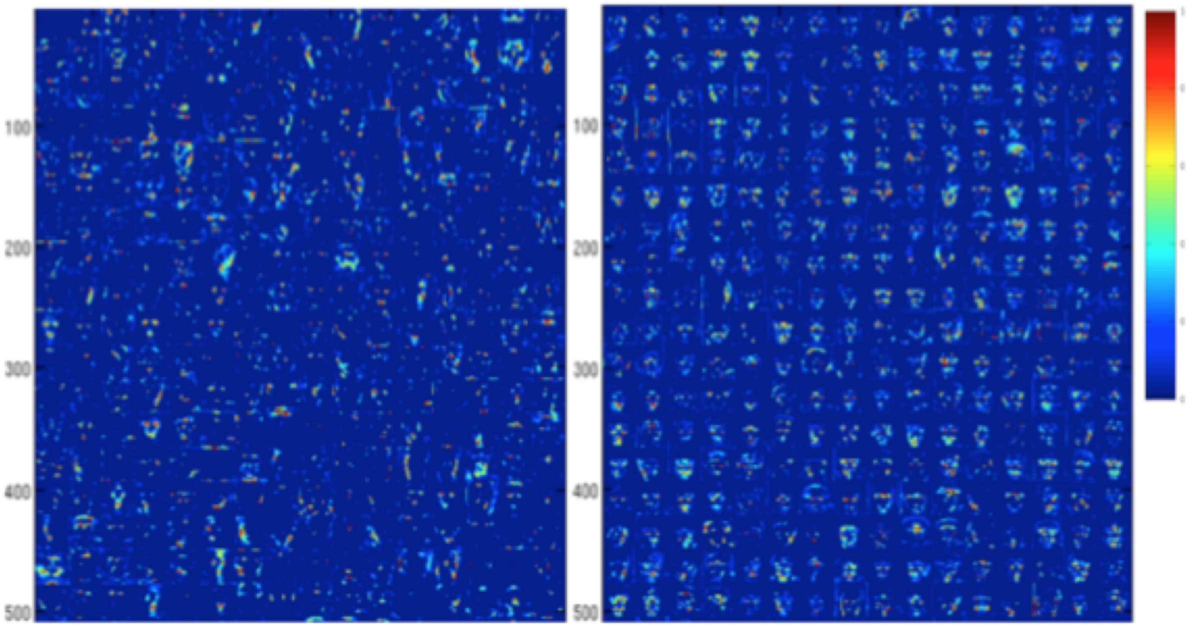


Figure 13.1: Visualization of selected feature maps from Group 4, with FVGG only (left) and with E-Net (right).

both F1 score and accuracy measures. Compared to the state of the art approaches, The EAC-Net shows improvement of F1 scores in all the AU detections results, except for AU2 (DRML) and AU15 (JPML), and of accuracy measures for all the 12 AUs. We can also see that the F1 score and accuracy measures all have improved from E-Net to EAC-Net: even though the cropping layers of the C-Net only slightly increases the average accuracy by 0.7%, the F1 score, which is a more appropriate indicator of the performance of the algorithm, increases by 3.5% over the E-Net. We know that the major role of the cropping layers is for a ”smart” alignment, so it might not have a significant effect on the faces of the BP4D data, which are mostly close to frontal view.

In evaluating our EAC-Net, we applied cross validation, in each round, we used the images of 23 out of the total 41 human subjects for training and the rest for testing. We hope by applying this cross validation, the trained model can focus more on detecting AUs rather than identifies of the human subjects. Due to the limited number of the subjects, identityinformation might also be learned. To reduce this influence, we could use the following two approaches: 1) Collecting more diverse data for training. If we have enough number of human subjects, the individual ID information will have less impact on AU recognition

Table 13.4: F1 score comparison of AU detection on DISFA dataset

AU	LSVM	APL[47]	DRML[47]	FVGG	E-Net	EAC
1	10.8	11.4	17.3	32.5	37.2	41.5
2	10.0	12.0	17.7	24.3	6.1	26.4
4	21.8	30.1	37.4	61.0	47.4	66.4
6	15.7	12.4	29.0	34.2	52.5	50.7
9	11.5	10.1	10.7	1.67	13.4	80.5
12	70.4	65.9	37.7	72.1	71.1	89.3
25	12.0	21.4	38.5	87.3	84.2	88.9
26	22.1	26.9	20.1	7.1	43.5	15.6
Avg	21.8	23.8	26.7	40.2	44.4	48.5

Table 13.5: Accuracy comparison of AU detection on DISFA dataset

AU	LSVM	APL[47]	DRML[47]	FVGG	E-Net	EAC
1	21.6	32.7	53.3	82.7	75.1	85.6
2	15.8	27.8	53.2	83.6	82.5	84.9
4	17.2	37.9	60.0	74.1	74.5	79.1
6	8.7	13.6	54.9	64.2	77.4	69.1
9	15.0	64.4	51.5	87.1	84.0	88.1
12	93.8	94.2	54.6	67.8	70.1	90.0
25	3.4	50.4	45.6	78.6	73.8	80.5
26	20.1	47.1	45.3	61.7	68.6	64.8
Avg	27.5	46.0	52.3	74.9	75.7	80.6

results. 2) Redesigning the model loss function by adding a regularization part, for example, by using the ID loss as an reverse loss in order to have an impact on the model loss, so that the model training process will learn more on AU detection and less on the ID information of each subject.

13.2.2 Results on DISFA dataset

We follow the setting in [47] to evaluate our approach on the DISFA dataset. Since the DISFA dataset is smaller in diversity and has less positive samples of AUs than BP4D, we do not directly train the model on DISFA. Instead, we use the trained model from BP4D. In DISFA, only 8 AUs are labeled; 3 of them do not exist in BP4D. Therefore we use our pretrained fine-tuned FVGG, E-Net and EAC Net to extract features from the DISFA images. Afterward,

Table 13.6: AU occurrence rates in DISFA dataset

AU1	AU2	AU4	AU6	AU9	AU12	AU25	AU26
4.9%	4.3%	15.2%	7.8%	4.1%	12.88%	27.7%	8.8%

we use linear regression to transform our 1x2048 features to 1x8 AU prediction labels. 27 subjects are split into 3 folds to make sure the predictions are independent.

The AU detection accuracy and F1 score are shown in Tables 13.4 and 13.5. Compared to the state of the art approaches, we see more significant improvement than that with BP4D. The F1 score of the EAC-Net increases by 4.1% over the E-Net, and the average accuracy increases by 4.9%. More importantly, the improvement yielded by C-Net is more significant than by the E-Net. Note that the significant improvement on DISFA is due to the following reasons:

(1) Our proposed approach is capable of balancing the training datasets. The DISFA dataset is more imbalanced than BP4D. If we directly use all the raw data, the AU occurrence rates of AU 4, 12, and 25 are much higher than the others (AU 1, 2, 6, 9, and 26), as shown in Table 13.6. Our preprocessing in balancing the data can improve the AU detection results.

(2) Our approach is more robust in dealing with wild images. The DISFA subjects have small rotation angles to the frontal view, so normalization is usually required in most the state-of-the-art approaches, while our approach only needs to know the approximate landmarks positions on the faces. This makes our approach much more robust dealing with faces which are not in frontal view. This will be further shown in Section 13.4 on AU detection on face images with large head poses.

13.3 EAC-Net with partially occluded faces

The AU detection on partially occluded faces is a very challenging problem. Without complete faces, which could be occluded by hands, glasses, hats, and other people, traditional landmark or patch-based approaches may fail to detect AUs. A few researchers [56] [57] have tried to learn partial facial regions to in order to detect AUs. Since our EAC-Net has both region enhancing and region-based CNN models, we would like to test if our model can have a good performance in AU detection when the faces are partially occluded.

There are three main reasons for us to conduct this experiment. First of all, our model learned enhanced features and cropped local features in the enhancing and cropping layers of EAC-Net. After the cropping layers, the local convolutional features are merged and learned as a combined feature, therefore we believe during this process, some interesting correlation can be learned. Thus even without raw input data for some of the regions, our EAC-Net might still be able to detect the remaining AUs on a partially occluded face. Second, if the AUs are detectable from partial faces, it is necessary to extend our investigation to which parts of a face convey more information than others for AU detection. Third, we are also interested in knowing if it is possible to predict all AUs without the primary parts of the face visible for some AUs. If this is true, this will imply that the AUs mainly perceivable from the occluded parts might be guessed based on the learned AU correlations in our model.

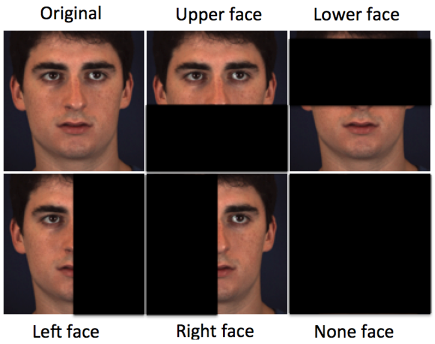


Figure 13.2: Occluded faces for AU detection.

Given above reasons, we conduct the following experiment on AU detection with occluded faces. As shown in Figure 13.2, we used our EAC-Net model directly to test on the BP4D faces where only upper, lower, left and right half-face are visible, respectively. By investigating the model performance, we would like to know which parts of the faces are more important in determining AU states. To make sure the results is better than a random guess, we also ran the test with the test data where whole faces are occluded. For a fair comparison with our previous AU detection results using the EAC-Net on original face images in BP4D, the same settings with BP4D training are used. In our experiment, we not only detect the "perceivable" AUs on the occluded faces but also try to predict the "unperceivable" ones, such as predicting the lower face AUs from the upper face images.

The occluded face AU detection results are shown in Table 13.7. In the table, "Lower" means only lower half of the face are used for testing, the same goes for "Upper", "Left", and "Right". For "None", we simply set all the pixel values to be 0s and feed the zero image into the EAC-Net. For columns Lower and Upper, since some AUs cannot be seen when running such a test on occluded faces, we use underlined numbers to note certain AUs results are "guessed" by the EAC-Net model. Some interesting conclusions can be made based on the results. Here are several observations:

(1) We can predict the "unseen" AUs status with partial faces. Comparing our "unseen" AU predictions (underlined numbers in Table 13.7) to a totally unavailable result, we can see

Table 13.7: F1 score comparison of AU detection on partially occluded faces of BP4D dataset

AU	Lower	Upper	Right	Left	None	EAC
1	31.8	27.4	31.4	25.2	27.7	**39.0**
2	30.0	31.6	34.6	32.4	4.5	**35.2**
4	21.8	29.1	21.1	28.7	22.8	**48.6**
6	70.5	54.9	39.7	52.9	64.8	**76.1**
7	72.3	**74.4**	66.4	70.1	58.6	72.9
10	77.0	64.6	60.9	62.6	52.6	**81.9**
12	75.0	67.6	57.9	59.9	56.9	**86.2**
14	58.5	51.6	48.2	45.7	44.5	**58.8**
15	15.0	14.8	7.3	18.4	3.8	**37.5**
17	58.1	39.3	38.7	37.8	6.0	**59.1**
23	28.6	18.9	27.1	12.5	13.14	**35.9**
24	16.3	13.0	4.3	7.6	4.9	**35.8**
Avg	46.3	40.6	36.5	37.8	29.8	**55.9**

that the EAC-Net always provides better-than-random-guess results with the four kinds of half occluded face images, although there are obvious differences among the four cases.

(2) It seems that the lower half of a face is the most useful part for AU detection. If we only look into the lower part of each face for AU detection, the EAC-Net model can produce comparable results to the DRML[47] and FVGG result and yield much better performance than the one from the other three parts (i.e., upper, left, and right). Even for some unseen AUs, like AU1(Inner brow raiser), AU2 (outer brow raiser), AU4 (Brow lower), the results are similar to or better than the results generated from “Up” face where these areas are seen.

(3) We also notice that the face AU correlation is important in single AU detection. For some obviously independent AUs like AU7 (Lid tighter), the appearance features may be enough and the “visibility” of that part can produce the best result (74% for AU7 in Upper faces). But for AUs like brow related and lip related ones, they are more dependent on the occurrence of some other AUs. For example, results could be different on detection of AU17 and AU 23 with the lower part or entire face being used. Including the upper face information can help infer whether the AUs of ”chin raiser” and ”lip tighter” are active.

(4) The upper/lower half-faces overall can generate better results than left/right half-faces, which means the proposed EAC-Net might not learn the symmetrical relationship of the face very well. This is an interesting finding that inspires us to design a model with such a function in our future work.

(5) From Table 13.7 we can find that the predictions results of some of the “unseen” AUs

Table 13.8: Pitch and yaw angles for the 9 poses, and the detected frames using pEAC-Net from the validation (development) dataset (For each pose, there are around 11,000 images)

Poses	Angles (yaw, pitch) in degrees	Detected frames: %
1	(-40, -40)	66.87
2	(-40, -20)	98.18
3	(-40, 0)	99.99
4	(0, -40)	99.96
5	(0, -20)	100
6	(0, 0)	100
7	(40, -40)	88.67
8	(40, -20)	98.14
9	(40, 0)	99.46

using the lower half-faces are even better than the "seen" AUs' in upper half-faces. This is probably because during the training process, not only the direct AU features, but also the relationship of different AUs are learned. Our deep learning model always try to yield an overall designed loss; this is also the limitation of deep learning. If we want the deep learning to have a better and also more balanced results, we may need to design more sophisticated structures and loss functions to cover more situations.

From the occluded faces experiment, we can see the proposed EAC-Net can predict reliable AU results with only half faces, especially with lower half face images. This may be a good sign for some scenes where only lower faces can be seen, such that the upper faces are occluded by sun glasses and hats. Knowing the correlation of AUs plays an important role in AU detection for design better AU detection models.

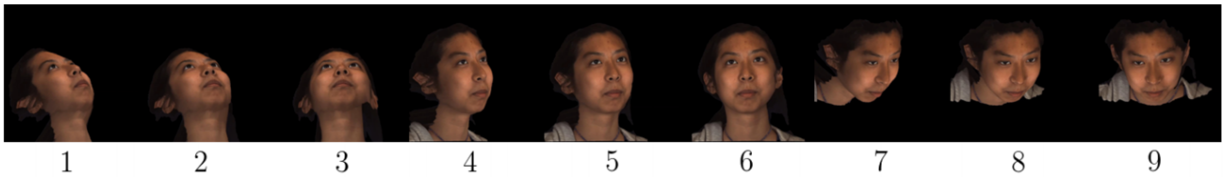


Figure 13.3: Nine head poses of a face in the BP4D AU dataset with large head poses from FERA 2017 data[63].

Table 13.9: AU detection results on BP4D images with large head poses: all 9 poses

	FERA Baseline		Our pEAC-Net	
AU	F1	Accuracy	F1	Accuracy
1	15.4	57.0	27.2	90.3
4	17.2	52.0	33.2	88.3
6	56.4	67.6	69.9	85.3
7	72.7	64.2	80.8	85.4
10	69.2	63.8	83.4	88.3
12	64.7	66.0	80.2	88.2
14	62.2	62.2	62.1	78.6
15	14.6	30.7	25.1	90.7
17	22.4	48.5	34.2	78.2
23	20.7	37.3	26.1	89.1
Avg	41.6	54.9	52.2	86.3

13.4 EAC-Net on faces with large head poses

In our previous experiments, we have found that our EAC-Net not only exhibits excellent performance in near frontal static face images for AUs detection, but also its ability in dealing with AUs detection and prediction on partially occluded facial images. However, in real-world scenarios, it is rare for people to show their spontaneous expressions without changing head poses. In our EAC-Net algorithm design (Chapter 11), we have hypothesized that our approach is adaptive to facial poses since we have created the attention map and the local convolutional networks are based on facial landmarks. To verify whether the proposed approach can really predict AUs with large head poses, we evaluate our algorithm on a newly derived dataset with different views of face images, which were generated from the BP4D database as the FERA 2017 challenge dataset [63].

The dataset used in the FERA 2017 challenge was generated from BP4D and BP4D+ [64]. BP4D+ is an expanded version of BP4D with more subjects in multi-modality. In our previous experiments, we employed the 2D version of BP4D to train the E-Net and EAC-Net. In fact, the BP4D and BP4D+ have also captured a 3D face model for each face image frame. In the FERA 2017 challenge, in order to create an AU dataset with multiple poses, the 3D face model of each face is rotated by pitch angles of -40, -20, and 0 degrees, and yaw angles of-40, 0, and 40 degrees, respectively, from its frontal pose, so in total there are nine poses for each face in total. Figure 13.3 shows one example, and Table 13.8 lists the pitch and yaw angles of each of the nine poses. As a result, nine videos with corresponding nine head poses are generated for each video sequence of a subject from the BP4D dataset.

Table 13.10: Comparison of F1 scores between the FERA baseline and our pEAC-Net on faces with the 9 poses

	FERA Baseline								
AU	1	2	3	4	5	6	7	8	9
1	10.3	15.0	13.6	19.3	19.6	17.1	18.0	14.5	12.3
4	15.0	15.9	14.8	19.1	19.0	18.3	20.2	20.2	17.5
6	50.5	55.7	13.4	68.9	74.7	72.4	56.0	53.2	49.3
7	72.1	72.9	41.3	74.6	79.7	78.7	71.6	74.7	75.8
10	55.4	71.0	64.2	77.7	77.6	75.0	63.9	67.9	65.9
12	52.2	67.8	18.4	78.6	80.9	77.1	59.6	63.8	60.1
14	51.5	56.3	9.0	67.5	72.4	74.4	61.9	67.0	67.8
15	13.1	15.0	14.2	14.6	14.6	14.6	14.3	15.9	15.2
17	17.3	25.1	23.5	24.2	24.6	24.1	22.0	21.1	19.5
23	22.7	22.9	19.9	20.8	19.6	16.6	20.1	20.1	20.8
Avg	36.0	41.8	23.2	46.5	48.2	46.8	40.8	41.8	40.4
	Our pEAC-Net								
AU	1	2	3	4	5	6	7	8	9
1	18.8	37.5	22.2	10.1	40.0	66.7	40.1	33.3	40.0
4	7.0	40.0	33.3	40.0	30.7	66.7	25.0	18.2	8.0
6	66.7	66.7	72.2	60.8	68.2	80.0	69.2	75.0	52.1
7	66.7	62.9	73.1	75.7	85.7	85.7	83.7	84.7	73.3
10	85.0	69.2	76.9	83.3	90.9	75.0	62.8	92.3	82.8
12	74.3	68.5	85.1	74.2	86.7	66.7	64.3	84.2	72.0
14	66.7	51.1	51.2	68.9	59.3	50.0	60.0	61.2	74.1
15	57.1	5.1	36.4	25.0	28.6	10.2	7.0	40.0	11.8
17	23.5	36.4	34.7	35.3	46.2	50.0	58.8	21.4	44.4
23	44.4	36.3	7.3	57.1	47.1	6.3	9.0	46.2	11.7
Avg	50.3	46.9	48.5	52.1	58.3	54.1	46.4	55.7	46.2

Table 13.11: Comparison of accuracy scores between the FERA baseline and our pEAC-Net on faces with the 9 poses

	FERA Baseline								
AU	1	2	3	4	5	6	7	8	9
1	25.2	35.9	32.9	56.3	58.0	50.0	72.9	90.2	91.5
4	30.0	37.2	14.3	45.6	55.4	62.7	50.1	81.0	92.0
6	73.2	72.7	69.6	81.3	82.9	79.1	66.7	47.3	35.7
7	60.6	64.2	53.9	69.5	73.8	72.8	61.0	60.4	61.7
10	56.3	65.0	71.1	74.9	72.2	67.5	61.2	56.0	50.3
12	62.7	70.7	60.9	81.0	81.1	76.4	63.6	53.9	43.6
14	61.8	62.6	51.6	71.2	73.0	73.0	60.8	53.3	52.4
15	39.8	21.8	10.2	67.8	21.5	12.1	36.8	28.5	37.8
17	69.9	46.8	24.6	74.1	59.8	52.2	46.4	30.6	32.3
23	27.1	27.5	55.4	50.8	46.6	59.4	34.9	18.9	15.4
Avg	50.7	50.5	44.4	67.3	62.4	60.5	55.4	52.0	51.3
	Our pEAC-Net								
AU	1	2	3	4	5	6	7	8	9
1	82.0	80.0	72.0	88.0	82.0	98.0	94.0	92.0	88.0
4	90.0	82.0	84.0	94.0	82.0	94.0	88.0	82.0	70.0
6	82.0	80.0	80.0	82.0	72.0	94.0	84.0	76.0	78.0
7	74.0	60.0	72.0	82.0	80.0	98.0	86.0	82.0	84.0
10	88.0	68.0	76.0	88.0	88.0	96.0	74.0	90.0	90.0
12	82.0	78.0	86.0	82.0	84.0	96.0	80.0	82.0	86.0
14	76.0	54.0	58.0	82.0	56.00	98.0	76.0	62.0	86.0
15	94.0	88.0	86.0	88.0	90.0	94	92.0	88.0	70.0
17	74.0	58.0	70.0	78.0	72.0	88.0	86.0	56.0	70.0
23	90.0	86.0	80.0	94.0	82.0	94.0	90.0	86.0	70.0
Avg	83.2	73.4	76.4	85.8	78.8	95.0	85.0	79.6	79.2

Table 13.12: AU detection results on BP4D faces with large head poses using pEAC-Net with a face alignment pre-processing

	F1 Score								
AU	1	2	3	4	5	6	7	8	9
1	19.3	13.9	13.3	6.8	25.0	31.3	18.2	0	0
4	25.0	0	33.3	14.2	25.0	42.1	6.1	14.3	37.8
6	58.3	60.0	65.0	73.5	74.1	63.1	69.1	51.1	45.7
7	80.4	76.7	73.9	88.8	80.5	78.9	87.4	82.7	70.4
10	83.7	78.9	72.2	89.2	85.3	86.1	85.7	76.5	77.8
12	72.4	79.3	73.1	85.3	78.6	84.7	73.1	65.5	74.1
14	36.4	29.2	48.0	30.4	42.6	33.3	44.4	53.1	31.8
15	31.3	0	22.2	33.3	11.11	28.6	25.6	27.8	16.0
17	30.2	21.8	34.6	18.5	49.1	27.5	28.6	37.9	17.8
23	22.2	10.0	10.5	40.0	33.3	37.5	6.9	34.1	7.7
Avg	45.9	37.0	44.6	48.1	50.5	51.3	44.5	44.3	37.6
	Accuracy								
AU	1	2	3	4	5	6	7	8	9
1	50.0	26.0	22.0	46.0	40.0	56.0	82.0	70.0	60.0
4	52.0	62.0	60.0	52.0	64.0	78.0	38.0	52.0	70.0
6	60.0	68.0	72.0	64.0	72.0	72.0	66.0	54.0	62.0
7	68.0	66.0	62.0	80.0	70.0	68.0	78.0	74.0	58.0
10	76.0	68.0	60.0	82.0	78.0	80.0	78.0	68.0	68.0
12	68.0	76.0	66.0	78.0	74.0	82.0	66.0	60.0	70.0
14	44.0	42.0	48.0	36.0	46.0	44.0	40.0	54.0	38.0
15	56.0	78.0	72.0	60.0	68.0	70.0	42.0	48.0	56.0
17	26.0	14.0	32.0	12.0	38.0	26.0	20.0	28.0	26.0
23	58.0	62.0	66.0	58.0	52.0	60.0	46.0	46.0	52.0
Avg	55.8	56.2	56.0	56.8	60.2	63.6	55.6	55.4	56.0

In our experiment, we train our EAC-Net by extracting the provided videos with 9 poses from the FERA 2017 training set. We note the EAC model trained on face images with large head poses as *pEAC-Net*. The number of images for training is about 260,000, and similar balancing strategy as we used in Table 13.1 is applied for boosting up the numbers of samples of the under-represented AUs. We use the same training configurations as we did in frontal view BP4D AU detection.

For comparison purpose, the FERA 2017 organizers [63] have provided AU detection baseline results on both its validation and test datasets, which are both from BP4D+. However, at the moment, only the performance results of the baseline approach on the validation (development) dataset is available to researchers. Therefore we only compare the results of our EAC approach with the baseline results on the validation dataset. We also use the landmarks provided by the challenge organizers, so the configurations of the dataset for testing on the validation dataset are the same with the baseline approach.

To further investigate whether our model can avoid face alignment by directly applying AU detection to face images with large head poses, we have also aligned 1,000 randomly selected images from each of the 9 poses. We applied the pEAC-Net trained on face images with large head poses to the aligned face images in an attempt to verify if the alignment would improve or degrade the AU detection performance.

The overall F1 and accuracy scores on all the nine poses are reported in Table 13.9, against the results of the baseline approach on the FERA development dataset. Table 13.10 and Table 13.11 show the F1 scores and accuracy values on face images of the nine individual poses, respectively. To make it easier to compare, the baseline results for all the 9 poses on the validation dataset are also listed. Table 13.12 lists the F1 and accuracy results of AU detection with aligned BP4D face images within 9 pose views.

From the results, we have several interesting observations as follows:

(1) Based on the pEAC-Net AU detection results on the BP4D face images with large head poses, the pEAC-Net shows its ability to detect AUs from facial images with different poses. Comparing to the baseline approach, there is a 10.6% improvement in F1 score. Although it is not fair to compare this with the EAC-Net results on the frontal view data of BP4D, we can see the performance improvement is at the same level. This shows that the pEAC-Net is able to detect AUs on face images with large head poses.

(2) The dataset of BP4D face images with large head poses provides 9 fixed view angles. We test our pEAC-Net on the 9 poses and find that in all the 9 angles, the pEAC-Net can make a better AU prediction than the baseline approach. Same to the finding from the baseline approach, the best single view result that our pEAC-Net has achieved is also on the pose #5, which is the F1-score at 58.3% for the 20-degree pitch angle from the frontal view.

This is interesting and might be due to the appearance of facial expressions when viewed from 20-degree angle being better than the frontal view in 3D space, while the face is not largely occluded. Note that this finding is also similar or compatible to the finding for non-frontal view facial expression classification reported in [65], where the similar performance was also achieved for 30-degree view as compared to the other view angles in recognizing six prototypic facial expressions from the static 3DFE database [66].

(3) Another observation we can make is that with larger angle orientations, the performance is generally becoming worse due to more information loss. However, our pEAC-Net performs relatively much more robust to view angle changes: the lowest F1 scores of our pEAC-Net are comparable to the highest F1 scores with the baseline approach.

(4) The aligned face AU detection experiment gives us many valuable insights into our pEAC-Net model. Our pEAC-Net can directly work on face images with large head poses and predict better AU results on all the nine poses than the results with an alignment preprocessing step. This shows the orientation invariant ability of our pEAC-Net model in dealing with head pose variations on AU detections, although large pose may still affect the AU detection due to information loss. As a comparison, alignment is not only time consuming (from only 30 ms to 2500 ms per image) but probably also degrades the fidelity of original facial expressions.

13.5 AU evaluation with temporal features

We showed that our proposed approach can achieve good performance in static image AU detection, we also mentioned that by employing temporal features, we can obtain more useful information, so we conduct experiments to see if the LSTM based fused feature can show better detection results. In our evaluation, we compute F1 scores for 12 AUs in BP4D and 8 AUs in DISFA. F1 scores can be compared directly as an indicator of the performance of different algorithms on each AU. The overall performance of the algorithm is described by the average F1 score.

13.5.1 Adaptive learning vs. conventional CNN

We proposed our ROI Nets for the adaptive region learning in 11. Compared to the conventional CNNs which share the same set of convolutional filters for the whole feature map, we hypothesize that by learning ROIs separately, a better understanding of AUs can be achieved. To validate this hypothesis, we train 2 neural networks on the BP4D dataset: a fine-tuned VGG model - FVGG, and the ROI Nets (on top of the basic VGG model). 12 AUs are used together, so the loss function is based on the predicted results for the 12 AUs. To prevent extreme loss explode which will stop the training, we added offsets to the loss function as shown by Equation 13.2, where l is the label and p is the generated probability for an AU.

$$Loss = -\Sigma(l \cdot \log(\frac{p+0.05}{1.05}) + (1-l) \cdot \log(\frac{1.05-p}{1.05})) \tag{13.2}$$

The two models are both based on static images. During each iteration, we randomly select 50 images as a batch to compute the training loss. SGD is employed for back propagation. The VGG net pretrained parameters are used for initialize the model, and the parameters of the first 8 convolutional layers are not updated during training. This makes the set of parameters smaller, which helps the training algorithm converge. To make the name more straightforward, we call the ”C-Net” ROI Nets. We use the proposed structure (VGG Net + ROI Nets) in Chapter 11 to train the adaptive region learning mode - which we still call ROI Nets. The new designed regional convolutional filters are initialized following a gaussian distribution. For the conventional fine-tuned VGG (FVGG) net, only the last prediction layer of the basic VGG model is replaced with a fully connected layer with 12 kernels. We use sigmoid activation functions for the 12 AU probability generators. The two deep models both start with the same learning rate 0.001 which is decreased when the loss is stable. Momentum for both models are set to 0.9.

The final models of both ROI Nets and FVGG are obtained after training the deep net 20,000 times. We then compare the F1 scores for each AU. The results are shown in Figure 13.4. We can see that region learning with ROI Nets yields significant improvement, on average by 12.4%.

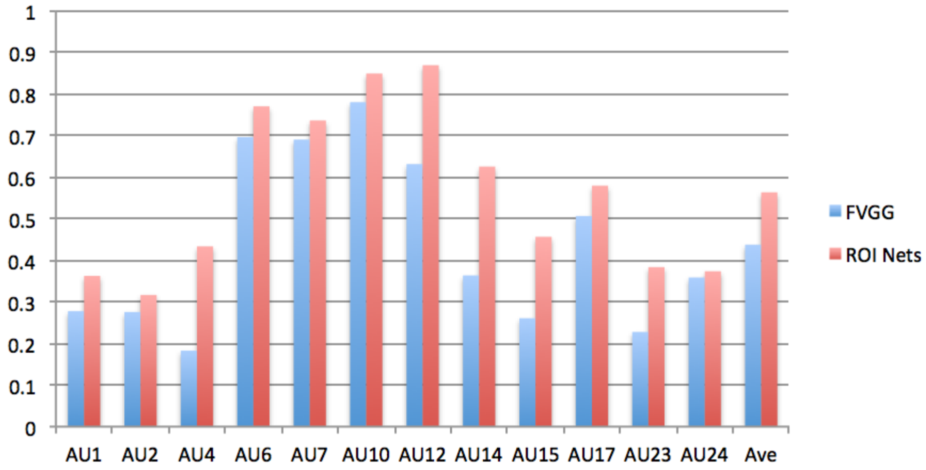


Figure 13.4: Comparison of FVGG and ROI-Nets in AU detection on BP4D

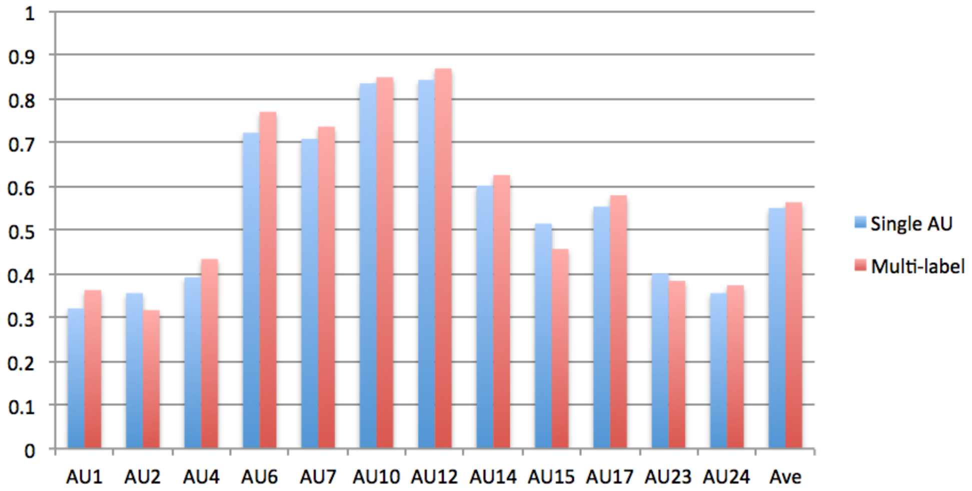


Figure 13.5: Comparison of single and multi-label learning on BP4D

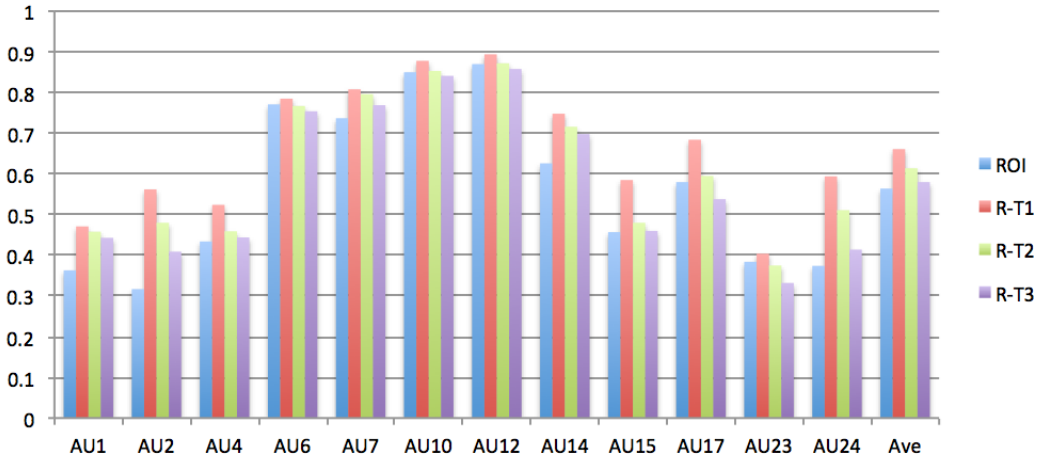


Figure 13.6: Comparison of static image and temporal fusion in AU detection on BP4D

13.5.2 Single vs. multi-label AU detection

In our proposed ROI Nets, the regions are determined based on the positions where the AUs take place. Since each AU has corresponding regions, we may use only the local learned features to represent the AU for detection. This single AU detection approach differs from the approach we use for the adaptive region learning evaluation (Figure 9.2) where we concatenate all the AUs features as one fused feature. Our hypothesis is, by concatenating multiple AU features, we may obtain valuable global information as a supplement for individual AU detection or to provide more correlations. However, it's also possible that it brings some noise to the "purity" of an AU feature. To validate our hypothesis, we conduct an experiment to compare single AU detection and multi-label AU detection. In multi-label AU detection, one image is labeled with multiple AUs. In this case, we cannot guarantee that we are able to provide the same number of positive and negative samples for all AUs. But for single AU detection, since the training for each AU is performed separately, we can prepare the training data for each AU in a way that the training data is always balanced during training. The AU detection results for single vs multiple AU detection is shown in Figure 13.5.

By comparison, we can clearly see that even with equal positive and negative sample distribution, the multi-label AU detection slightly outperforms the single AU detection approach in most AUs, on average by 1.3%. That implies that the global information does have an important impact on the fusion learning. We have some more interesting findings if we look into the different AU detection results. For the under-represented AUs (where the AU shows up less frequently in the dataset), such as AU2, AU15, AU23, the balancing of training samples (as in the single AU detection) can boost the performance more significantly. Whereas for some highly related AUs such as AU6 and AU12, both for happy, the multi-label learning has a higher chance to learn this correlation and improve the AU detection for these two AUs.

13.5.3 Temporal vs. static

A facial action always has a temporal component, hence knowing the previous state of a facial expression can definitely improve the AU detection. We proposed the LSTM layer for fusing the temporal information with static image features. From our previous evaluations, the best performance was obtained for static images with the ROI Nets. In this experiment, we use the ROI model as a baseline to compare with region cropping recurrent temporal model (noted as R-T in figures and tables). Here, the LSTM layers are used for fusing the static image features. 512 LSTM kernels are employed to construct each LSTM layer. We then utilize 24 frames as a sequence to represent the video. In our data preparation, we follow the same framework as the one we used to train the static image learning models. The only difference is

Table 13.13: F1 score on BP4D dataset (ROI: ROI Nets; R-Ti: ROI Nets + i-layer LSTM Net)

AU	LSVM	JPML[38]	DRML[47]	CPM[43]	CNN+LSTM[56]	FVGG	ROI	R-T1	R-T2
1	23.2	32.6	36.4	43.4	31.4	27.8	36.2	**47.1**	45.8
2	22.8	25.6	41.8	40.7	31.1	27.6	31.6	**56.2**	48.0
4	23.1	37.4	43.0	43.4	**71.4**	18.3	43.4	52.4	45.9
6	27.2	42.3	55.0	59.2	63.3	69.7	77.1	**78.5**	76.7
7	47.1	50.5	67.0	61.3	77.1	69.1	73.7	**80.8**	79.6
10	77.2	72.2	66.3	62.1	45.0	78.1	85.0	**87.8**	85.3
12	63.7	74.1	65.8	68.5	82.6	63.2	87.0	**89.4**	87.2
14	64.3	65.7	54.1	52.5	72.9	36.4	62.6	**74.8**	71.6
15	18.4	38.1	36.7	34.0	33.2	26.1	45.7	**58.5**	48.0
17	33.0	40.0	48.0	54.3	53.9	50.7	58.0	**68.4**	59.5
23	19.4	30.4	31.7	39.5	38.6	22.8	38.3	**40.4**	37.5
24	20.7	42.3	30.0	37.8	37.0	35.9	37.4	**59.4**	51.1
Avg	35.3	45.9	48.3	50.0	53.2	43.8	56.4	**66.1**	61.4

Table 13.14: F1 score on BP4D dataset (ROI: ROI Nets; R-Ti: ROI Nets + i-layer LSTM Net)

AU	LSVM	JPML	DRML	CPM	CNN+T1	E-Net	EAC	ROI	R-T1	R-T2
1	23.2	32.6	36.4	43.4	31.4	37.6	39.0	36.2	**47.1**	45.8
2	22.8	25.6	41.8	40.7	31.1	32.1	35.2	31.6	**56.2**	48.0
4	23.1	37.4	43.0	43.4	**71.4**	44.2	48.6	43.4	52.4	45.9
6	27.2	42.3	55.0	59.2	63.3	74.5	72.9	77.1	**78.5**	76.7
7	47.1	50.5	67.0	61.3	77.1	80.8	**81.9**	73.7	80.8	79.6
10	77.2	72.2	66.3	62.1	45.0	85.1	86.2	85.0	**87.8**	85.3
12	63.7	74.1	65.8	68.5	82.6	63.2	87.0	87.0	**89.4**	87.2
14	64.3	65.7	54.1	52.5	72.9	36.4	62.6	62.6	**74.8**	71.6
15	18.4	38.1	36.7	34.0	33.2	56.8	**58.8**	45.7	58.5	48.0
17	33.0	40.0	48.0	54.3	53.9	55.6	59.1	58.0	**68.4**	59.5
23	19.4	30.4	31.7	39.5	38.6	21.9	35.9	38.3	**40.4**	37.5
24	20.7	42.3	30.0	37.8	37.0	29.1	35.8	37.4	**59.4**	51.1
Avg	35.3	45.9	48.3	50.0	53.2	52.1	55.9	56.4	**66.1**	61.4

Table 13.15: F1 score on DISFA dataset

AU	LSVM	APL[47]	DRML[47]	FVGG	ROI	R-T1
1	10.8	11.4	17.3	32.5	41.5	**42.6**
2	10.0	12.0	17.7	24.3	26.4	**27.2**
4	21.8	30.1	37.4	61.0	**66.4**	65.5
6	15.7	12.4	29.0	34.2	50.7	**55.5**
9	11.5	10.1	10.7	1.67	8.5	**22.8**
12	70.4	65.9	37.7	72.1	**89.3**	82.9
25	12.0	21.4	38.5	87.3	**88.9**	88.3
26	22.1	26.9	20.1	7.1	15.6	**25.9**
Avg	21.8	23.8	26.7	40.2	48.5	**51.3**

that to construct the image sequence, we randomly find other 23 images prior to the selected image from the same subject. This will create more non-repeatable training data. Afterward, the sequence is fed into the training model. To find the best LSTM structure, we tried 1 (in R-T1), 2 (in R-T2) and 3 (in R-T3) stacked LSTM layers for AU detection, as demonstrated in Figure 12.2. The AU detection results are shown in Figure 13.6.

From the results shown in Figure 13.6, we can clearly observe the improvement in AU detection due to applying the LSTM layers. The average F1 score is also improved by 9.7% using R-T1 over ROI Nets. Another conclusion we can make here is that with more LSTM layers, the performance decreases, as the ROI features are sufficient to represent the AU images and one LSTM layer is enough to reveal the temporal corrections.

13.6 Performance comparison

By observing the results of our previous experiments, we can clearly see that the ROI Nets can learn more powerful local AU features that would result in better AU detection compared to conventional CNNs. The performance was similar in single AU detection and multi-label AU detection, but the multi-label detection approach shows slightly better overall performance due to the strong correlation among AUs and richer global information. In the static/temporal exploration experiment, we witnessed that the LSTM Net with one LSTM layer boosts the AU detection accuracy by a 9.7% average F1 improvement, which implies that the temporal context information plays a very important role in detecting facial actions.

To compare our approaches with other state of the art methods, we have collected the F1 measures of the most popular methods in same 3-fold settings based on BP4D in Table 13.14. The approaches includes a traditional SVM based method, a 2-D landmark feature based approach, JPML [38], the Confidence Preserving Machine (CPM) [43], a block-based region learning static CNN, DRML [47], and a recurrent net fusing LSTM with simple CNN,

CNN+LSTM [56]. For our proposed approaches, we first use the FVGG as the baseline approach. Then we show the results of adaptive ROI Nets based on static images. Finally, we test our ROI Nets + our LSTM based recurrent approach with one and two LSTM layers (RC+T1,RC+T2). All the results can be seen in Table 13.14. On average, our best model R-T1 achieves a 12.9% improvement compared to the state of the art approach. Across the 12 AUs, our R-T1 model outperforms the best in the literature except AU4, where CNN+LSTM performs the best.

To further explore the capabilities of our proposed approach, we run the comparison on DISFA dataset as well. Not as popular as BP4D, fewer state of the art approaches report their results on DISFA. We use the BP4D trained model to extract features from all the images in DISFA and conduct a 3-fold cross evaluation with the extracted features. For static image evaluation, we directly run multi-label linear regression and for temporal evaluation, we use the structure that shows the best performance in BP4D evaluation, that is, a one layer LSTM to train the DISFA temporal model. The results are shown in Table 13.15. As we can see, our R-T1 model leads to a 25% improvement over the state of the art model.

From the results in Tables 13.14 and 13.15, our proposed approaches have the best performance in both static and sequence image based AU detection. In the static images based AU detection using deep learning, our ROI Nets outperforms the state of the art deep learning approach, DRML. Our proposed adaptive region cropping method shares the same idea of learning different sets of convolutional filters for different sub-regions, but our method has the following advantages that makes it different from the state of the art:

1) Our sub region selection is adaptive. DRML used a straightforward image dividing strategy. Assuming the facial images are aligned, each image is equally divided into 8x8=64 sub-regions. This framework in easy to implement, but we need to make sure that the face images are actually aligned in the first place. In order to assure this precondition, all the faces need to be transformed to a neutral shape. This may cause information loss since the faces of different individuals may have different shapes or sizes. In addition, if the original faces are not in frontal pose, we may also lose some appearance features after changing the pose. On the contrary, we select the regions of interest adaptively. Our approach works based on the detected landmarks and the positions of facial action muscles, which are biologically meaningful. Also note that our approach is robust to landmark position errors. This is because the feature maps in our network go through several pooling layers. Imagine that the position detection error in original image of size 224x224 is 10 pixel. With the pooling layer for cropping the feature map being of size 14x14, the error turns to be less than 1 pixel. This significantly improves our proposed adaptive region cropping net.

2) A very deep pretrained network (VGG) is used as the base. DRML creates a shallow

convolutional network for region based AU detection. Instead of training everything from scratch, we choose to borrow parameters from an existing very deep CNN model. The main advantage of this approach is that the pretrained model has been trained with millions of images. Although the tasks are different, the parameters are transferable. With the pretrained model as the starting point of our AU detection training, we can achieve a more powerful model than by training a shallow neural network.

In sequential image based AU detection, Chu et al. [57] designed a network by combining both CNN and LSTM. To obtain the spatiotemporal fusion features, the last layer features of the CNN and LSTM nets are concatenated. Different from their use of AlexNet for static image feature extraction, we have proposed the adaptive region cropping convolutional net. We use LSTM to fuse the temporal deep features as well, but we have also compared different layers of LSTM and noticed that one layer LSTM shows the best performance.

Intuitively, more LSTM layers should yield higher recognition rates. But in our experiment, we have found we can achieve the best performance when we use one single LSTM layer rather than two or three layers. This might be related to the features fed into the LSTM net. We suspect that if the features are well learned by one layer of LSTM, that single LSTM layer is already sufficient for achieving the temporal fusion. Adding more layers may lead to more gradient loss and more parameters might also make the model hard to converge.

14 CIFE-AU: A New AU Dataset and Evaluation

14.1 CIFE-AU: a dataset for AU detection in the ”wild”

The CIFE dataset is an facial expression dataset collected from web. We choose the CIFE dataset as the base for creating an AU dataset – CIFE-AU – because there are much more subjects involved in the dataset compared to most of the AU datasets collected in laboratory settings. The images in CIFE are with different poses, lighting conditions and resolutions. This makes the dataset more challenging for classification but also provides more representative samples for deep learning.

In the Facial Action Coding System (FACS) [26], more than 40 AUs are listed, but not all these AUs are equally important. In BP4D, 12 AUs are labeled, whereas in DISFA only 8 AUs are labeled. In our new dataset CIFE-AU, we used the most popular 14 AUs, containing all the AUs from BP4D and two additional ones from DISFA. Here we mainly choose the configuration of BP4D because BP4D is a larger and more diverse dataset, making it more convenient for us to train a based deep model and conduct comparison. We added another 2 AUs since we think this 2 AUs is important in representing some facial expressions. These 14 AUs are listed in Table 14.1, with their names in the third column and their indices in FACS in the second column. Their presences in BP4D and DISFA are also noted in the table. We hope that these 14 AUs are sufficient to represent multiple facial expressions.

Table 14.1: List of the AUs in the CIFE-AU dataset

AU No.	FACS index	AU Name	Notes
0	1	Inner Brow Raiser	in BP4D & DISFA
1	2	Outer Brow Raiser	in BP4D & DISFA
2	4	Brow Lowerer	in BP4D & DISFA
3	6	Cheek Raiser	in BP4D & DISFA
4	7	Lid Tightener	in BP4D
5	10	Upper Lip Raiser	in BP4D
6	12	Lip Corner Puller	in BP4D & DISFA
7	14	Dimpler	in BP4D
8	15	Lip Corner Depressor	in BP4D
9	17	Chin Raiser	in BP4D
10	23	Lip Tightener	in BP4D
11	24	Lip Pressor	in BP4D
12	9	Nose Wrinkler	in DISFA
13	25	Lip Apart	in DISFA

For labeling AUs in a video sequence, we just need to specify the start and end frames of an AU in the sequence which means many frames can be skipped for labeling between the two frames. But, for labeling AUs for an image dataset, all image frames need to be examined to generate AU labels. This makes the AU labeling for such a dataset more challenging. In total, 14758x14 decisions have to be made for all the 14 AUs in the CIFE dataset with 14756 images. To make the work less tedious, an AU labeling interface is designed to help label the CIFE dataset. On the interface, the AU coders are asked to pick up the active samples that belong to one of the 14 active AU. Using the interface, an AU coder goes through AUs one after after another, and for each AU (whose No. can be input on top of the interface and whose description is also shown), the coder goes through all the samples, screen by screen, by simply clicking on the subwindow of an image that includes the AU to turn its background from blue to red. The coder can even come back from the place where he or she stops the last time. Figure 14.1 shows a screen in the selection of images for labeling AU No. 6 (FACS index is 12): Lip Corner Puller.

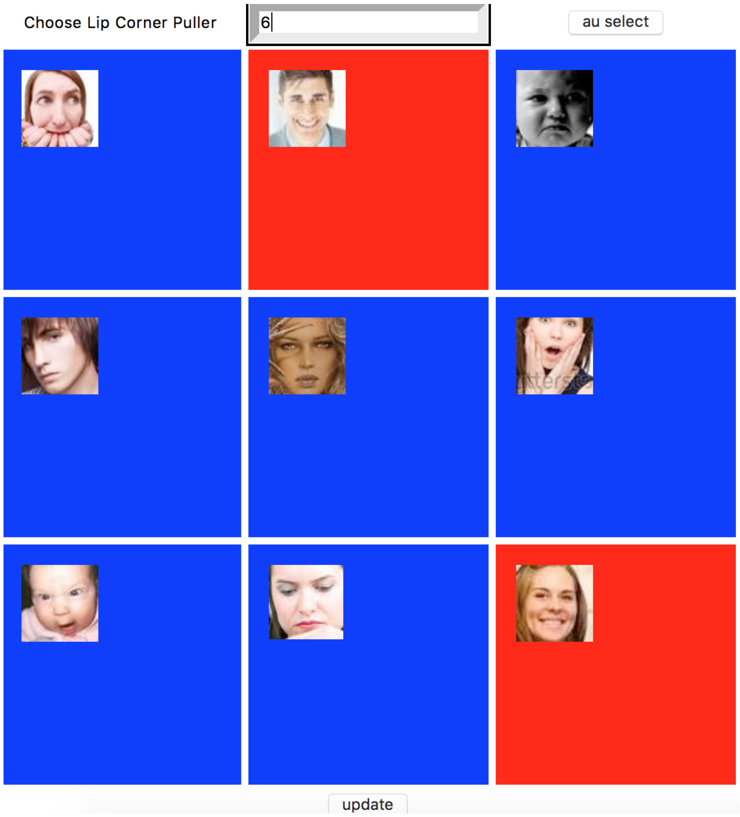

Choose Lip Corner Puller 6

Figure 14.1: An AU Labeling interface

In our work, 3 experienced AU coders were employed to label all the images. We then

Table 14.2: AU occurrence rates in the CIFE-AU dataset

AU Index	1	2	4	6	7	10	12	14	15	17	23	24	9	25
Occurence	9.2%	6.1%	7.2%	14.3%	19.5%	17.4%	18.9%	6.1%	6.5%	6.7%	36.7%	33.5%	13.2%	47.3%

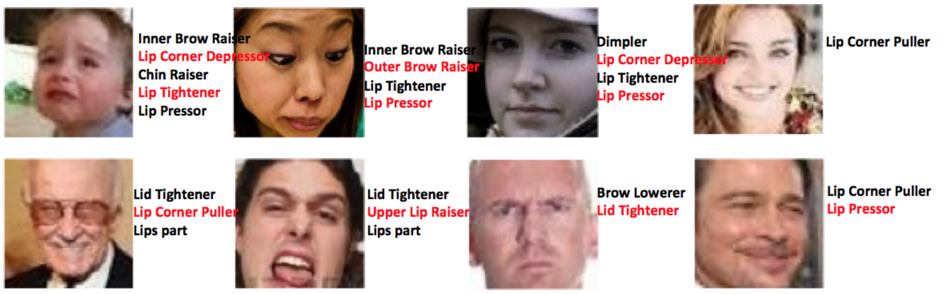


Figure 14.2: Examples of AU detection in the "wild"

merged the coded results and obtained the AU labels, leading to the CIFE-AU dataset.

14.2 Evaluation on CIFE-AU

The proposed E-Net shows its outstanding performance on standard AU detection when compared to the state of the art methods. But whether this approach is useful in real scene AU detection needs to be evaluated. This is also the motivation for us to generate the CIFE-AU dataset. The CIFE-AU dataset contains images from the wild, hence there are fewer restrictions on the expressions and AUs. This also makes the AU detection more challenging.

In the CIFE-AU dataset statistics, we notice that the occurrence rates for AUs are quite biased (Table 14.2). This makes it difficult to jointly train the deep model for all AUs. To ensure that sufficient image samples for both active and inactive AUs can be fed to the neural network during training, the training is performed on each AU individually. Each training batch contains 50 samples with half being active and the other half being inactive samples. In the original CIFE, the entire dataset is already divided into training and test parts, so we directly use these two sub-datasets for training and testing the model. For comparison purposes, we implement the state of the art approach DRML introduced in [47], therefore three models - DRML, FVGG and E-Net - are then compared against each other on both of their accuracy and F1 score values for all the 14 AUs. Based on the evaluation results (Table 14.3), the proposed E-Net is able to achieve the best performance: on average, 9.9% increase in the F1 score and 21.0% in accuracy over DRML.

Table 14.3 also gives a detailed comparison of accuracy and F1 score values for all the

Table 14.3: Results on the CIFE-AU dataset

	Accuracy			F1 score		
AU	DRML[47]	FVGG	E-Net	DRML[47]	FVGG	E-Net
1	47.3	77.3	**86.7**	19.2	31.2	**37.2**
2	62.5	87.9	**88.4**	16.8	24.9	2**5.8**
4	68.7	80.9	**87.4**	16.5	**19.5**	16.9
6	65.1	66.8	**76.4**	30.1	33.6	**36.6**
7	71.4	69.9	**73.6**	34.8	46.8	**47.2**
9	49.4	73.1	**79.6**	27.2	36.9	**37.0**
10	45.2	**64.3**	63.2	32.1	35.9	**39.0**
12	58.7	70.5	**78.8**	33.1	46.9	**50.1**
14	55.2	**86.8**	82.7	13.4	**18.2**	15.4
15	60.1	84.9	**85.8**	16.5	20.8	**22.1**
17	57.3	76.5	**87.9**	19.6	25.4	**31.2**
23	55.9	73.2	**73.5**	42.3	62.8	**65.6**
24	53.7	66.5	**73.8**	56.2	61.6	**64.0**
25	62.3	**71.0**	69.9	63.7	70.4	**72.3**
Avg	58.1	74.9	**79.1**	30.1	38.3	**40.0**

AUs using the CIFE-AU test dataset. Although we tried to balance the data before feeding it to the network during training, the lack of sufficient active samples still affects the under-represented AUs and the F1 scores for these AUs whose occurrence rates are very low. On the other hand, for the AUs with more active samples, the models show better performance both in accuracy and F1 scores. Despite these difficulties, we still see that compared to the state of the art approaches, the proposed E-Net can achieve better results on the "wild" AU dataset. Figure 14.2 shows examples of some AU prediction results on the test images. For the DRML approach, the input images need to be aligned in a way that similar facial regions fall into the same blocks. The images in CIFE-AU though are mostly random posed, which makes it hard for the state of the art approach to perform well. For E-Net on the other hand, the attention layers will move according to the landmark key points and even when the landmarks have localization errors, the E-Net can still handle the AU detection reasonably well.

15 AU Detection: Applications and Discussions

Action Units (AUs) are the basic facial movements that work as the building blocks in formularizing multiple facial expressions, such as attention monitoring [79]. pain detection [80], deception detection [81] and personality evaluation[76]. The successful detection of AUs will greatly facilitate the analysis of the complicated facial actions or expressions. AU detection has been studied for decades as one of the basic facial computing problems and many interesting approaches have been proposed.

15.1 Possible applications

Compared to facial expression recognition which is trying to classify facial image to 7 basic classes, AU detection focuses on more basic facial element actions. The combination of basic AU actions can lead to more interesting questions, so the AU detection can have more usage in real life applications.

There are many problems in the application side which needs reliable AU detection. One of the very important application is driver status monitoring in assistive driving[79]. Driver's attention is a very important factor to ensure the driving safety. Being able to detect the tiredness and distraction of the driver can help assistive driving system have better understanding of driving status. AU detection can give a good evaluation of tiredness and attention [79]. There are many other applications such as pain detection and deception detection that AU detection can provide important features.

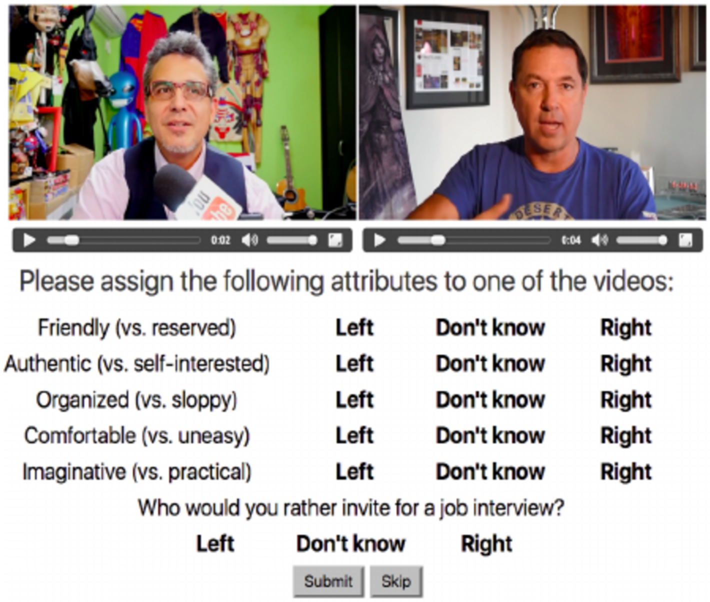
15.2 Personality evaluation: a real-world problem

15.2.1 Interview evaluation: problem statement

Personality is a strong predictor of important life outcomes like happiness and longevity, quality of relationships with peers, family, occupational choice, satisfaction, and performance, community involvement, criminal activity, and political ideology [76]. Personality plays an important role in the way people manage the images they convey in self-presentations and employment interviews, trying to affect the audience first impressions and increase effectiveness. As is known, the first impression made is highly important in many contexts, such as human resourcing or job interviews.

15.2.2 FI: the first impressions dataset and the task

The First Impressions (FI) dataset [76] include 10,000 video clips, each is 15-second long and is annotated with personality traits by AMT (Amazon Mechanical Turk) workers. The



Please assign the following attributes to one of the videos:

Friendly (vs. reserved)	Left	Don't know	Right
Authentic (vs. self-interested)	Left	Don't know	Right
Organized (vs. sloppy)	Left	Don't know	Right
Comfortable (vs. uneasy)	Left	Don't know	Right
Imaginative (vs. practical)	Left	Don't know	Right

Who would you rather invite for a job interview?

Left Don't know Right

Figure 15.1: Illuatration of AMT based first impression data labeling.

traits correspond to the "big five" personality traits used in psychology and are well known to hiring managers using standardized personality profiling: (1) extroversion, (2) agreeableness, (3) conscientiousness, (4) neuroticism, and (5) openness to experience. Here we call these five personalities as 5 traits.

Figure 15.1 illustrates a user interface for the generation of the 5 labels for the video clips. AMT workers are asked to compare 2 videos from the dataset instead of label all in order to minimize human bias. For each pair of video clips, five traits are compared by AMT workers: friendly vs. reserved for extroversion, authentic vs. self-interest for agreeableness, organized vs. sloppy for conscientiousness, comfortable vs. uneasy for neuroticism, and imaginative vs. practical for openness to experience. The workers just compare different pairs of video clips for the 5 traits. Each video clip may also be compared with many other other different video clips. So all the scores are comparison scores (0s or 1s) between pairs of videos. These scores are used to generate a cardinal score (ranging from 0 to 1)for each video clip by fitting a BTL model [77]. And the generated cardinal score is regarded as the ground truth for that video clip in the first impression dataset. Figure 15.2 shows several samples labels obtained through this crowd source labeling approach. We can see in figure 15.2, the labeled scores for 5 traits represent the level of different personality profiles. The emotion states can give a clear clue for us to understand the personality of the candidates and the score is able to describe the difference. But for algorithm designs, the size of face and the environment seem changing a

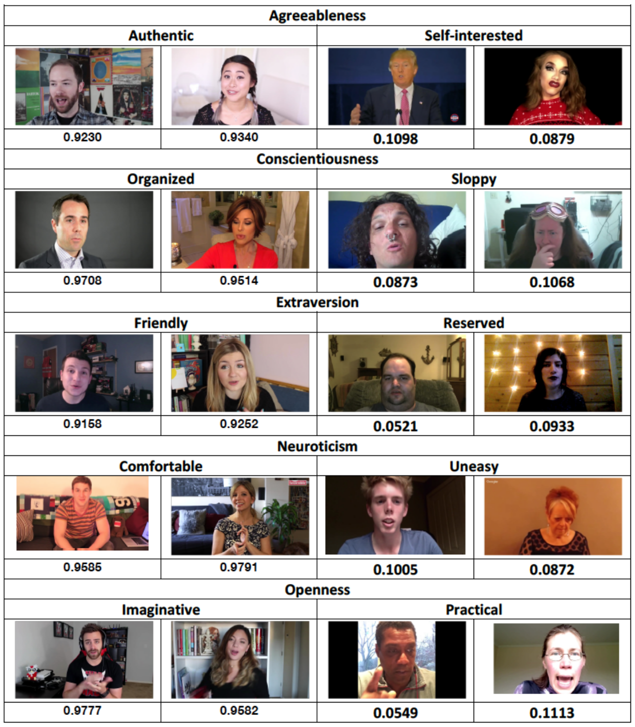


Figure 15.2: Samples of labeled videos sequence.

lot across different videos, poses are also varying a lot, so this is a very good dataset to test the robustness of our AU and emotion detection algorithms.

The First Impression dataset is a typical video based supervised learning task. The task is to use deep learning to regressed to 5 values represent the "big five" personality traits. We can apply our facial computing approaches to this task. If our proposed approach can show the good performance in standard and candid AU dataset, it should be able to provide some good performance in personality evaluation task.

15.3 Discussion

Starting from the fine-tuned VGG to the EAC-Net to the LSTM-Net, we see a steady improvement in AU detection performance. From the experimental comparisons, we can see our proposed EAC-Net leads to significant improvement over the state-of-the-art baselines in AU detection, since the enhancing and cropping layer make the neural network have a better understanding about the AU features. LSTM based temporal feature fusion with the LSTM-Net shows the capability to find AU relationships in nearby frames and hence have boosted the AU detection performance. We obtained better results on the BP4D AU datasets compared to the state of the art approaches. Furthermore, by observing the comparison results of the LSTM-Net and the ROI-Net (a simplified EAC-Net but more effective and efficient) on BP4D, we witnessed that the LSTM Net with one LSTM layer boosts the AU detection accuracy by a 9.7% average F1 improvement, which implies that the temporal context information plays a very important role in detecting facial actions. On the DISFA dataset, the LSTM-Net outperforms the stat-of-the-art approaches by an even larger margin, a 25% improvement. To extend our work to more "wild" scenarios, we generated a new dataset by manually labeling the CIFE dataset. This makes our CIFE-AU dataset different from the existing AU datasets in the sense that the CIFE-AU contains labeled web images without constraints in poses, lighting conditions, and environments. Our approach shows its robustness in detecting AUs in "wild" images. We will look forward to testing our approaches in more real applications like personality evaluation in the next step.

16 Conclusions and Future Work

16.1 Concluding Remarks

In this thesis, the facial computing problem is tackled in two frontiers: facial expression recognition as a higher level task and action unit detection as a lower level task. We use deep learning models for both tasks, and three aspects are considered in undertaking the tasks: data, algorithms, and applications.

In the first part, we propose a recursive framework in order to achieve real scene facial expression recognition . We first build a candid image facial expression dataset –CIFE– by parsing Web expression images from image search engines. The CNN-based deep learning approaches are then employed to train robust facial expression predictors, while fine-tuning approaches are also constructed to improve facial expression accuracy. To collect real scene images, we have designed a facial expression interaction game based on our deep learning model that was trained with the CIFE dataset. With users playing both the general and the customized versions of the face game, the correctly labeled facial expression images are selected and saved, which help us build the GaMo dataset. We also have run a self-evaluation of the quality of the data labeling and proposed a self-cleaning mechanism to improve the quality of the data. To validate the effectiveness of our framework, we compared GaMo and CIFE based their balancedness, recognition accuracy, the effectiveness of using strictly balanced subsets, the impact of data-cleansing, and feedback from human subjects. The experiments show that our framework can build a reliable facial expression predictor for real scenes.

In the AU detection part, we design the enhancing net (E-Net) to force the neural network to pay more attention to AU interest regions on face images. We have also proposed the cropping net (C-Net) or region of interest (ROI Net) to ensure that individual networks learn features in “aligned” facial areas. This makes our EAC-Net – the integration of the E-Net and C-Net – more robust to facial shifts and orientation differences. We have evaluated our proposed E-Net and EAC-Net against the state-of-the-art approaches. The AU detection results show that our approach can achieve better performance on commonly used AU datasets. With deep pretrained models and a “smarter” way to focus on interest regions, the proposed approach shows its power in AU detection on multiple datasets including BP4D and DISFA. We have further tested our EAC-Net on BP4D occluded face data and found that our learning model is able to detect, and to a certain extent, predict AUs even with half faces visible, especially for lower-half faces. We have also shown that the proposed EAC-Net works well with faces with large pose differences.

Furthermore, we looked into the integration of three effective mechanisms: the region adaption learning, temporal fusion and single/multi-label AU learning, in AU detection and proposed a novel approach for the integration. We first used our adaptive region of interest cropping net (the C-Net or ROI Net), which compared to conventional CNN, proves to be able to learn separate filters for different regions and can improve the accuracy of AU detection. We then analyzed the proposed model by training it in a multi-label AU detection manner and showed that the new model can outperform a single AU detection model. We finally explored the LSTM-based temporal fusion approach, which boosted the AU detection performance significantly compared to static image-based approaches. We also tried to find an optimal structure of LSTM layers to connect with the proposed ROI nets to achieve the best results for AU detection.

16.2 Limitations and Some Future Directions

Deep learning models do have some limitations, which also have been relfected during the course of this thesis research.

First, our deep learning models, as most of the practices in the field, always try to control an overall designed loss. Statistically this may yield a high overall performance in classification or recognition, but may not so for some specific classes, especially some underrepresented classes. Therefore, if we want the deep learning to have a better and also more balanced results, we may need to design more sophisticated structures and loss functions to cover more situations.

Second, the internal structures of the deep models are still hard to explain. Overall, our best understanding so far is that the earlier layers extract lower level features, and latter layers higher level features, but the exact mechanisms are usually hard to explain. We have tried to look into the feature maps extracted from the expression recognition models and the AU detection models, and we have seen some more meaningful feature maps with our fine-tuning and attention structures, but these findings are hard to be generalized.

Third, intuitively, more layers should lead to better recognition results. We have seen this in our facial expression recognition part, in which using VGG with more layers led to better recognition results. We have also hypothesized that more LSTM layers should yield higher recognition rates. But in our experiment, we have found that we can achieve the best performance when we use one single LSTM layer rather than two or three layers. This might be related to the features fed into the LSTM net. We suspect that if the features are well learned by one layer of LSTM, that single LSTM layer is already sufficient for achieving the temporal fusion. Adding more layers may lead to more gradient loss and more parameters might also make the model hard to converge.

Finally, it is the general thought of how far we can go with deep facial models with the current achievements in performance. In the future, we may extend our facial expression recognition and AU detection work in many aspects. Our recursive framework with the game-based approach can help us obtain much more real-world data and provide more robust models for facial expression recognition in real life. We would also like to improve our work in AU detection to find more responsive areas for the enhancing and cropping nets rather than manually locating the positions at present. In addition, we may explore integrating more temporal information into the EAC-Net framework to deal with the wild video AU detection problem.

17 List of Candidate's Publications During PhD Study

W. Li, F. Abtahi, Z. Zhu, L. Yin. EAC-Net: Deep Nets with Enhancing and Cropping for Facial Action Unit Detection. (Extended version of FG accepted paper) Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence, March 2017.

W. Li, F. Abtahi, Z. Zhu, Action Unit Detection with Region Adaptation, Multi-labeling Learning and Optimal Temporal Fusing. 2017 Conference on Computer Vision and Pattern Recognition (CVPR 2017).

W. Li, F. Abtahi, Z. Zhu, L. Yin. EAC-Net: A Region-based Deep Enhancing and Cropping Approach for Facial Action Unit Detection. The 12th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2017).

C. Tsangouri, W. Li, Z. Zhu, F. Abtahi and T. Ro, An Interactive Facial-Expression Training Platform for Individuals with Autism Spectrum Disorder, 2016 IEEE MIT Undergraduate Research Technology Conference (URTC), November 4-6, 2016 at MIT, Cambridge USA (Tsanouri is an undergraduate student directly mentored by the candidate. This paper was among the top 6 nominations for Best Paper Award from the reviewers, and CT's oral presentation was voted Best Presentation in the machine learning/cloud computing track from the audience)

W. Li, F. Abtahi, C. Tsangouri, Z Zhu.A Game-based Framework for Building an "In-the-Wild" facial expression Dataset. Computer Vision and Pattern Recognition Workshops (CVPRW), 2016 IEEE Conference on. IEEE, 2016.

W Li, C Tsangouri, F Abtahi, Z Zhu. A Recursive Framework for Expression Recognition: From Web Images to Deep Models to Game Dataset. Journal of Machine Vision and Application. Under review.

W. Li, Z Su, M. Li, Z Zhu, A Deep-Learning Approach to Facial Expression Recognition with Candid Images. In Machine Vision Applications (MVA), 2015 14th IAPR International Conference on, pp. 279-282. IEEE, 2015.

W. Li, F Abtahi, Z. A Deep Feature based Multi-kernel Learning Approach for Video facial expression Recognition. In Proceedings of the 2015 ACM on International Conference on

Multimodal Interaction, pp. 483-490. ACM, 2015 (Top 5 in terms of performance among 74 submissions).

F. Abhati, W. Li, Z Zhu, T. Ro, Multimodal Speaker Recognition using Deep Belief Networks, Computer Vision and Pattern Recognition Workshops (CVPRW), 2015 IEEE Conference on. IEEE, 2015.

W. Li, M. Goldberg, X. Li, Z Zhu, Face Recognition by 3D Registration for the Visually Impaired Using a RGB-D Sensor, In European Conference on Computer Vision, pp. 763-777. Springer International Publishing, 2014.

References

[1] Xie, L., Hong, R., Zhang, B., Tian, Q. (2015, June). Image classification and retrieval are one. In Proceedings of the 5th ACM on International Conference on Multimedia Retrieval (pp. 3-10). ACM.

[2] Mavadati, S. M., Mahoor, M. H., Bartlett, K., Trinh, P., Cohn, J. F. (2013). Disfa: A spontaneous facial action intensity database. Affective Computing, IEEE Transactions on, 4(2), 151-160.

[3] Kanade, T., Cohn, J. F., Tian, Y. (2000). Comprehensive database for facial expression analysis. In Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on (pp. 46-53). IEEE.

[4] Cohn, J. F., Ambadar, Z., Ekman, P. (2007). Observer-based measurement of facial expression with the Facial Action Coding System. The handbook of expression elicitation and assessment, 203-221.

[5] Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., Matthews, I. (2010, June). The extended cohn-kanade dataset (ck+): A complete dataset for action unit and expression-specified expression. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on (pp. 94-101). IEEE.

[6] Shan, C., Gong, S., McOwan, P. W. (2009). Facial expression recognition based on local binary patterns: A comprehensive study. Image and Vision Computing, 27(6), 803-816.

[7] Li, W., Li, M., Su, Z., Zhu, Z. (2015, May). A deep-learning approach to facial expression recognition with candid images. In Machine Vision Applications (MVA), 2015 14th IAPR International Conference on (pp. 279-282). IEEE.

[8] Li, W., Abtahi, F., Zhu, Z. (2015, November). A deep feature based multi-kernel learning approach for video expression recognition. In Proceedings of the 2015 ACM on International Conference on Multimodal Interaction (pp. 483-490). ACM.

[9] Li, W., Farnaz A.,Tsangouri, C., Zhu Z.(2016). Towards an ”in-the-wild” facial expression dataset using a game-based framework. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 75-83).

[10] Pantic, M., Valstar, M., Rademaker, R., Maat, L. (2005, July). Web-based database for facial expression analysis. In Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on (pp. 5-pp). IEEE.

[11] Xiao, R., Zhao, Q., Zhang, D., Shi, P. (2011). Facial expression recognition on multiple manifolds. Pattern Recognition, 44(1), 107-116.

[12] Wang, Z., Wang, S., Ji, Q. (2013). Capturing complex spatio-temporal relations among facial muscles for facial expression recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3422-3429).

[13] Sikka, K., Dhall, A., Bartlett, M. (2015). Exemplar hidden Markov models for classification of facial expressions in videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 18-25).

[14] Taigman, Y., Yang, M., Ranzato, M. A., Wolf, L. (2014). Deepface: closing the gap to human-level performance in face verification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1701-1708).

[15] Sun, Y., Wang, X., Tang, X. (2014). Deep learning face representation from predicting 10,000 classes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1891-1898).

[16] Liu, P., Han, S., Meng, Z., Tong, Y. (2014). Facial expression recognition via a boosted deep belief network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1805-1812).

[17] Kim, Y., Lee, H., Provost, E. M. (2013, May). Deep learning for robust feature generation in audiovisual expression recognition. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on (pp. 3687-3691). IEEE.

[18] Jung, H., Lee, S., Yim, J., Park, S., Kim, J. (2015). Joint fine-tuning in deep neural networks for facial expression recognition. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2983-2991).

[19] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., Fei-Fei, L. (2009, June). Imagenet: a large-scale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on (pp. 248-255). IEEE.

[20] Von Ahn, L., Dabbish, L. (2004, April). Labeling images with a computer game. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 319-326). ACM.

[21] Mouro, A., Magalhes, J. (2013, October). Competitive affective gaming: winning with a smile. In Proceedings of the 21st ACM international conference on Multimedia (pp. 83-92). ACM. Chicago

[22] Deriso, D., Susskind, J., Krieger, L., Bartlett, M. (2012, October). expression mirror: a novel intervention for autism based on real-time expression recognition. In Computer Vision-ECCV 2012. Workshops and Demonstrations (pp. 671-674). Springer Berlin Heidelberg.

[23] Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1440-1448).

[24] Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[25] Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

[26] Ekman, Paul, and Erika L. Rosenberg. "What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)". Oxford University Press, USA, 1997.

[27] Girshick, Ross. "Fast r-cnn." In Proceedings of the IEEE International Conference on Computer Vision, pp. 1440-1448. 2015.

[28] Li, Wei, Farnaz Abtahi, and Zhigang Zhu. "A Deep Feature based Multi-kernel Learning Approach for Video facial expression Recognition." In Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, pp. 483-490. ACM, 2015.

[29] Yang, Zichao, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. "Stacked attention networks for image question answering." arXiv preprint arXiv:1511.02274 (2015).

[30] Zhao, Rui, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. "Saliency detection by multi-context deep learning." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1265-1274. 2015.

[31] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Spatial pyramid pooling in deep convolutional networks for visual recognition." In European Conference on Computer Vision, pp. 346-361. Springer International Publishing, 2014.

[32] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: Towards real-time object detection with region proposal networks." In Advances in neural information processing systems, pp. 91-99. 2015.

[33] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[34] Valstar, Michel, and Maja Pantic. "Fully automatic facial action unit detection and temporal analysis." In 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06), pp. 149-149. IEEE, 2006.

[35] Eleftheriadis, Stefanos, Ognjen Rudovic, and Maja Pantic. "Multi-conditional Latent Variable Model for Joint Facial Action Unit Detection." In Proceedings of the IEEE International Conference on Computer Vision, pp. 3792-3800. 2015.

[36] S. Koelstra, M. Pantic, and I. Y. Patras. A dynamic texturebased approach to recognition of facial actions and their temporal models. IEEE Transactions onPattern Analysis and Machine Intelligence, 32(11):1940-1954, 2010.

[37] Fabian Benitez-Quiroz, C., Ramprakash Srinivasan, and Aleix M. Martinez. "facial expressionet: An Accurate, Real-Time Algorithm for the Automatic Annotation of a Million Facial Expressions in the Wild." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[38] Zhao, Kaili, Wen-Sheng Chu, Fernando De la Torre, Jeffrey F. Cohn, and Honggang Zhang. "Joint patch and multi-label learning for facial action unit detection." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2207-2216. 2015.

[39] Wang, Ziheng, Yongqiang Li, Shangfei Wang, and Qiang Ji. "Capturing global semantic relationships for facial action unit recognition." In Proceedings of the IEEE International Conference on Computer Vision, pp. 3304-3311. 2013.

[40] Jaiswal, Shashank, and Michel Valstar. "Deep learning the dynamic appearance and shape of facial action units." In Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on, pp. 1-8. IEEE, 2016.

[41] Chu, Wen-Sheng, Fernando De la Torre, and Jeffery F. Cohn. "Selective transfer machine for personalized facial action unit detection." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3515-3522. 2013.

[42] Ding, Xiaoyu, Wen-Sheng Chu, Fernando De la Torre, Jeffery F. Cohn, and Qiao Wang. "Facial action unit event detection by cascade of tasks." In Proceedings of the IEEE International Conference on Computer Vision, pp. 2400-2407. 2013.

[43] Zeng, Jiabei, Wen-Sheng Chu, Fernando De la Torre, Jeffrey F. Cohn, and Zhang Xiong. "Confidence preserving machine for facial action unit detection." In Proceedings of the IEEE International Conference on Computer Vision, pp. 3622-3630. 2015.

[44] Wu, Yue, and Qiang Ji. "Constrained joint cascade regression framework for simultaneous facial action unit recognition and facial landmark detection." In Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on. 2016.

[45] Girard, Jeffrey M., Jeffrey F. Cohn, Laszlo A. Jeni, Simon Lucey, and Fernando De la Torre. "How much training data for facial action unit detection?" In Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on, vol. 1, pp. 1-8. IEEE, 2015.

[46] Gehrig, Tobias, Ziad Al-Halah, Hazam Kemal Ekenel, and Rainer Stiefelhagen. "Action unit intensity estimation using hierarchical partial least squares." In Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on, vol. 1, pp. 1-6. IEEE, 2015.

[47] Zhao, Kaili, Wen-Sheng Chu, and Honggang Zhang. "Deep Region and Multi-Label Learning for Facial Action Unit Detection." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3391-3399. 2016.

[48] Song, Yale, Daniel McDuff, Deepak Vasisht, and Ashish Kapoor. "Exploiting sparsity and co-occurrence structure for action unit recognition." In Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on, vol. 1, pp. 1-8. IEEE, 2015.

[49] Liu, Mengyi, Shaoxin Li, Shiguang Shan, and Xilin Chen. "Au-aware deep networks for facial expression recognition." In Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on, pp. 1-6. IEEE, 2013.

[50] Kazemi, Vahid, and Josephine Sullivan. "One millisecond face alignment with an ensemble of regression trees." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1867-1874. 2014.

[51] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." arXiv preprint arXiv:1512.03385 (2015).

[52] Zhang, Xing, Lijun Yin, Jeffrey F. Cohn, Shaun Canavan, Michael Reale, Andy Horowitz, Peng Liu, and Jeffrey M. Girard. "BP4D-Spontaneous: a high-resolution spontaneous 3D

dynamic facial expression database." Image and Vision Computing. Vol 32, no. 10 (2014): 692-706.

[53] Lucey, Patrick, Jeffrey F. Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and facial expression-specified expression." In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, pp. 94-101. IEEE, 2010.

[54] Mavadati, S. Mohammad, Mohammad H. Mahoor, Kevin Bartlett, Philip Trinh, and Jeffrey F. Cohn. "Disfa: A spontaneous facial action intensity database." IEEE Transactions on Affective Computing 4, no. 2 (2013): 151-160.

[55] Valstar, Michel F., Timur Almaev, Jeffrey M. Girard, Gary McKeown, Marc Mehu, Lijun Yin, Maja Pantic, and Jeffrey F. Cohn. "Fera 2015-second facial expression recognition and analysis challenge." In Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on, vol. 6, pp. 1-8. IEEE, 2015.

[56] Yang, Shuo, Ping Luo, Chen-Change Loy, and Xiaoou Tang. "From facial parts responses to face detection: A deep learning approach." In Proceedings of the IEEE International Conference on Computer Vision, pp. 3676-3684. 2015.

[57] Grafsgaard, Joseph, Joseph B. Wiggins, Kristy Elizabeth Boyer, Eric N. Wiebe, and James Lester. "Automatically recognizing facial expression: Predicting engagement and frustration." In Educational Data Mining 2013.

[58] Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-9. 2015.

[59] Ma, Shugao, Leonid Sigal, and Stan Sclaroff. "Learning activity progression in LSTMs for activity detection and early detection." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1942-1950. 2016.

[60] Graves, Alex, and Jrgen Schmidhuber. "Framewise phoneme classification with bidirectional LSTM and other neural network architectures." Neural Networks 18.5 (2005): 602-610.

[61] Christopher, Olah. "Understanding LSTM networks." colah's blog. August 27, 2015.

[62] Taigman, Y., Yang, M., Ranzato, M. A., Wolf, L. (2014). Deepface: closing the gap to human-level performance in face verification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1701-1708).

[63] Valstar, Michel F., Enrique Sanchez-Lozano, Jeffrey F. Cohn, Laszlo A. Jeni, Jeffrey M. Girard, Zheng Zhang, Lijun Yin, and Maja Pantic.“FERA 2017-Addressing Head Pose in the Third Facial Expression Recognition and Analysis Challenge.” arXiv preprint arXiv:1702.04174 (2017).

[64] Zhang, Zheng, Jeff M. Girard, Yue Wu, Xing Zhang, Peng Liu, Umur Ciftci, Shaun Canavan,Michael Reale, Andy Horowitz, Huiyuan Yang, Jeff F. Cohn, Qiang Ji, and Lijun Yin, ”Multimodal spontaneous facial expression corpus for human behavior analysis.” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3438-3446. 2016.

[65] Yuxiao Hu, Zhihong Zeng, Lijun Yin, Xiaozhou Wei, Xi Zhou, and Thomas S. Huang, ”Multi-view facial expression recognition”, The 8th International Conference on Automatic Face and Gesture Recognition (FGR08), Sept. 2008.

[66] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and Matt Rosato, ”A 3D facial expression database for facial behavior research”, IEEE 7th International Conference on Automatic Face and Gesture Recognition (FG06), April 2006, p211-216.

[67] Behrmann, Marlene, Galia Avidan, Grace Lee Leonard, Rutie Kimchi, Beatriz Luna, Kate Humphreys, and Nancy Minshew. “Configural processing in autism and its relationship to face processing.” Neuropsychologia 44, no. 1 (2006): 110-129.

[68] McDuff D, Kaliouby R, Senechal T, Amr M, Cohn J, Picard R. “Affectiva-mit facial expression dataset (am-fed): Naturalistic and spontaneous facial expressions collected.” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops 2013 (pp. 881-888).

[69] Cockburn, Jeff, Marni Bartlett, James Tanaka, Javier Movellan, Matt Pierce, and Robert Schultz. “Smilemaze: A tutoring system in real-time facial expression perception and production in children with autism spectrum disorder.” In ECAG 2008 Workshop Facial and Bodily Expressions for Control and Adaptation of Games, p. 3. 2008.

[70] Tanaka, James W., Julie M. Wolf, Cheryl Klaiman, Kathleen Koenig, Jeffrey Cockburn, Lauren Herlihy, Carla Brown, Sherin Stahl, Martha D. Kaiser, and Robert T. Schultz.

"Using computerized games to teach face recognition skills to children with autism spectrum disorder: the Let?s Face It! program." Journal of Child Psychology and Psychiatry 51, no. 8 (2010): 944-952.

[71] Golan, Ofer, Emma Ashwin, Yael Granader, Suzy McClintock, Kate Day, Victoria Leggett, and Simon Baron-Cohen. "Enhancing facial expression recognition in children with autism spectrum conditions: An intervention using animated vehicles with real facial expressional faces." Journal of autism and developmental disorders 40, no. 3 (2010): 269-279.

[72] Golan, Ofer, and Simon Baron-Cohen. "Systemizing empathy: Teaching adults with Asperger syndrome or high-functioning autism to recognize complex facial expressions using interactive multimedia." Development and psychopathology 18, no. 02 (2006): 591-617.

[73] Moore, Monique, and Sandra Calvert. "Brief report: Vocabulary acquisition for children with autism: Teacher or computer instruction." Journal of autism and developmental disorders 30, no. 4 (2000): 359-362.

[74] Heimann, Mikael, Keith E. Nelson, Tomas Tjus, and Christopher Gillberg. "Increasing reading and communication skills in children with autism through an interactive multimedia computer program." Journal of autism and developmental disorders 25, no. 5 (1995): 459-480.

[75] Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." In Advances in Neural Information Processing Systems, pp. 2672-2680. 2014.

[76] Ponce-Lopez, Victor, et al. "ChaLearn LAP 2016: First Round Challenge on First Impressions-Dataset and Results." ECCV 2016 Workshops. Springer International Publishing, 2016.

[77] Bradley, Ralph Allan, and Milton E. Terry. "Rank analysis of incomplete block designs: I. The method of paired comparisons." Biometrika 39.3/4 (1952): 324-345.

[78] Kazemi, Vahid, and Josephine Sullivan. "One millisecond face alignment with an ensemble of regression trees." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1867-1874. 2014.

[79] Rezaei, Mahdi, and Reinhard Klette. ”Look at the driver, look at the road: No distraction! No accident!.” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 129-136. 2014.

[80] Khan, Rizwan Ahmed, Alexandre Meyer, Hubert Konik, and Saida Bouakaz. ”Pain detection through shape and appearance features.” In Multimedia and Expo (ICME), 2013 IEEE International Conference on, pp. 1-6. IEEE, 2013.

[81] Granhag, Pr Anders, Aldert Vrij, and Bruno Verschuere. “Detecting deception: current challenges and cognitive approaches.” John Wiley, Sons, 2015.