FYP17019

Mak Tin Shing

3035187857

Lo Wang Kin

3035186401

Facial Expression Recognition

using Region-based Convolutional Neural Network

Interim Report

January 21, 2018

# Abstract

Facial Expression has always been an important role in face-to-face communication. Messages sent and received using facial expression are easily understood by human without much effort, but it is not an easy task for computers. Understanding human emotions through facial expression is still a challenge for computer system [1].

This project aims at improving facial expression recognition using the techniques of R-CNN. The method chosen is to extract critical area for classification before feeding them to a CNN. Two FER systems are included in the project, a baseline model using CNN, and a two-staged system using R-CNN as area extractor and CNN as classifier.

# Acknowledgements

I would like to express my gratitude towards Dr. K. P. Chan, supervisor of this project, for providing valuable advices as well as equipment for this project.

Contents

[Abstract 2](#_Toc504038708)

[Acknowledgements 3](#_Toc504038709)

[List of Figures 5](#_Toc504038710)

[List of Abbreviations 6](#_Toc504038711)

[1 Introduction 7](#_Toc504038712)

[2 Objective 8](#_Toc504038713)

[3 Methodology 9](#_Toc504038714)

[3.1 Hardware 9](#_Toc504038715)

[3.2 Framework 9](#_Toc504038716)

[3.3 Dataset 10](#_Toc504038717)

[3.4 FER systems for analysis 11](#_Toc504038718)

[3.4.1 System 1: Basic CNN 11](#_Toc504038719)

[3.4.2 System 2: R-CNN with modified CNN 11](#_Toc504038720)

[3.5 Merging Strategies 12](#_Toc504038721)

[3.5.1 Treating the image of regions as different channel of one image. 12](#_Toc504038722)

[3.5.2 Arranging the component linearly, feed as one image 14](#_Toc504038723)

[3.5.3 Arranging the component based on relative position, feed as one image 14](#_Toc504038724)

[3.5.4 Filling unextracted area with black 14](#_Toc504038725)

[3.6 Evaluation method 15](#_Toc504038726)

[4 State of Work 16](#_Toc504038727)

[4.1 Phase 1: Setup phase 16](#_Toc504038728)

[4.2 Phase 2: Experiment phase 16](#_Toc504038729)

[5 Difficulties encountered 17](#_Toc504038730)

[5.1 Limited information from the data set provided. 17](#_Toc504038731)

[5.2 Extraction of facial landmarks from the dataset 18](#_Toc504038732)

[5.3 High error rate of Faster R-CNN model 19](#_Toc504038733)

[6 Future Work Schedule 20](#_Toc504038734)

[7 Conclusion 21](#_Toc504038735)

[8 Reference 22](#_Toc504038736)

# List of Figures

[Figure 3.1 An example of a sequence in the dataset 10](#_Toc504039791)

[Figure 3.2 Basic CNN 11](#_Toc504039792)

[Figure 3.3 Network of the basic CNN 11](#_Toc504039793)

[Figure 3.4 R-CNN + modified CNN 12](#_Toc504039794)

[Figure 3.5 Normal image input 13](#_Toc504039795)

[Figure 3.6 Merging strategy 1 13](#_Toc504039796)

[Figure 3.7 Merging strategy 2 14](#_Toc504039797)

[Figure 3.8 Merging strategy 3 14](#_Toc504039798)

[Figure 3.9 Merging strategy 4 14](#_Toc504039799)

[Figure 5.1 Examples of images which cannot be labelled with any emotions. 17](#_Toc504039800)

[Figure 5.2 Visualization of boundary of left eye before(left) and after(right) boundary relax 18](#_Toc504039801)

[Figure 5.3 AP before and after relaxing boundary, note that some AP before relaxing boundary is too small to visualize 19](#_Toc504039802)

[Figure 5.4 Left eye detection by R-CNN with confidence > 0.5 19](#_Toc504039803)

# List of Abbreviations

|  |  |
| --- | --- |
| AP | Average Precision |
| CNN | Convolutional Neural Network |
| FER | Facial Expression Recognition  |
| GPU | Graphics Processing Unit |
| mAP | Mean Average Precision |
| R-CNN | Region-based Convolutional Neural Network  |
| VOC | Visual Object Classes |
| VOC 2012 | Visual Object Classes Challenge 2012 |
|  |  |

# Introduction

To facilitate a more natural communication between humans and computer, considerable progress has been made in the field of FER, especially in feature extraction algorithms and classification techniques in the past few years. Much development involves using a technique called Facial Action Coding System (FACS) [2,3]. FACS is a method that classifies different facial components, such as eyes and lips, into some Action Units (AUs). Combining different AUs, the computer can then recognize the expression. Some other methods involving classifiers like Naive Bayes classifiers and hidden Markov models [4].

The recent success in Convolutional Neural Network (CNN) has drawn much attention in the field of Facial Expression Recognition (FER). Different contributors try to apply modifications on CNN such as applying transformation on images at train time, using a decision tree for combining results of different neural networks, and use different pooling layers [5,6,7]. However, most CNN models usually cannot achieve both high recognition rate and high accuracy. Much experiments and research are needed to examine the success and drawbacks of different techniques.

To extend the use of CNN to object detection, the Region-based Convolutional Neural Networks (R-CNN) was developed in 2014 [8]. The algorithm was further optimized to Fast-R-CNN, Faster-R-CNN and Mask-R-CNN. R-CNN successfully improves the precision on PASCAL VOC2012 by 30%, compared to CNN [8]. The success of R-CNN could be a key to further improve the performance of FER.

This report will first discuss the objective of the project, and then carry on with the design and implementation details. After that, the current status, difficulties encountered, and remaining work schedule will be discussed.

# Objective

This goal of this project is to experiment the effectiveness of R-CNN on facial expression recognition. This project will start with implementing a functional FER system that uses traditional CNN as a baseline for comparison. Then a system that can extract facial component using R-CNN, and then classify facial expression using CNN, will be built. Different merging strategies of the facial components will be experimented. The performance of different techniques will be compared and analyzed.

# Methodology

## Hardware

A remote desktop under Ubuntu environment, provided by Dr. K.P. Chan, will be used as the main platform for development. The GPU of this device will be used for training process due to its better performance.

## Framework

This project uses Caffe framework for development. Caffe is chosen because of its high popularity among neural network, especially CNN researchers which makes finding pre-trained model and troubleshooting technical issues easier. Moreover, the main part of the project involves modifying and testing R-CNN, of which source code are written in Caffe. So, choosing Caffe greatly reduce the risk of translating.

While Caffe provide interface for both Python and MATLAB, Python is chosen in this project as the team have more experience on the language. Also, MATLAB is not provided in the remote machine provided and is not free of charge.

## Dataset

The FER dataset for training and testing is from Extended Cohn-Kanade Dataset (CK+) [9]. The dataset contains 593 sequences of images, each of them begins with a neutral expression and proceed to a peak emotion (see Figure 3.1). There is a total of 11424 images with 8 kinds of expressions, namely neutral, angry, contempt, disgust, fear, happy, sadness, and surprise. The dataset is chosen for its relatively high resolution, which is ideal for facial component extraction in the later stage.



Figure 3.1 An example of a sequence in the dataset

The first 20% of each sequence will be labeled as a neutral expression. Also for each sequence, the last 40% will be labeled as a peek emotion. The middle 40% is discarded since their expressions are too vague, which can be either neutral or peek emotion.

75% of the sequences will be used as the training dataset. 12.5% of the sequences will be reserved for validation, and the remaining 12.5% will be the testing dataset.

## FER systems for analysis

### System 1: Basic CNN

This will provide the baselines of the experiment. The network will be a basic CNN that classify image of facial expressions into emotions (figure 3.2). The network will train with and without transfer learning. The structure of the network will be a modified version of the BVLC Reference Caffenet[10]. The transfer learning version will use the pre-trained caffemodel of the BVLC Reference Caffenet from GitHub which trained with ImageNet dataset for 310,000 iterations. Figure 3.3 shows the structure of the network this system is using.



Figure 3.2 Basic CNN

Input (227x227x3)

Convolution (55x55x96)

ReLU

MAX-Pool (27x27x96)

Normalization

Convolution (27x27x256)

ReLU

MAX-Pool (13x13x256)

Normalization

Convolution (13x13x384)

ReLU

Convolution (13x13x384)

ReLU

Convolution (13x13x256)

ReLU

MAX-Pool (6x6x256)

Fully Connected (4096)

ReLU

Dropout

Fully Connected (4096)

ReLU

Dropout

Fully Connected (8)

Softmax

Figure 3.3 Network of the basic CNN

### System 2: R-CNN with modified CNN

This will be the focus of the project. R-CNN is an object detection using CNN. After trained, R-CNN can report the bound and label for objects found in each image. Although there is other facial component extraction method that not requiring training, R-CNN is choose as it is more versatile as new component can be extracted by adding training data.

The R-CNN part will be trained with custom data to extract eye, mouth and eyebrow from the original dataset, the modified CNN will then take those areas as input for both training and classification (figure 3.4).



Figure 3.4 R-CNN + modified CNN

## Merging Strategies

System 2 require feeding multiple image areas into classifying CNN. As traditional CNNs only take one image for training and classifying, a merging strategy is need. Below are the proposed merging strategies:

### Treating the image of regions as different channel of one image.

When passing image into CNN, normally, the image is represented as a 3D array of C × width × height. Where C is the number of channels of the image, which are often RGB channels (see figure 3.5). This strategy will involve changing C into number of regional images and feed the set of images as if they are different channel of one image (see figure 3.6). The width and height will be changed to max of each among images. The extra space in smaller image will be filled with black.



Figure 3.5 Normal image input



Figure 3.6 Merging strategy 1

### Arranging the component linearly, feed as one image

This strategy will arrange the component both horizontally and vertically and feed the one image for classification (figure 3.7).

 

Figure 3.7 Merging strategy 2

### Arranging the component based on relative position, feed as one image

This strategy involves removing the unbounded area from the image, while keeping the image in rectangular shape (figure 3.8).



Figure 3.8 Merging strategy 3

### Filling unextracted area with black

This strategy involves blacked the unextracted area (figure 3.8), this will be the baseline for designs (figure 3.9).



Figure 3.9 Merging strategy 4

## Evaluation method

System 1 and System 2 with different merging strategy will be train and test with the corresponding data. The highest mAP will be recorded for accuracy evaluation. mAP per training iteration will be compared for learning rate evaluation. The result can be break down into AP of different emotions for further analysis. Finally, if time allows, the systems will be tested will data of partial face and non-human face without additional training for versatility evaluation.

# State of Work

This project consists of two phases:

1. Setup phase
2. Experiment phase

## Phase 1: Setup phase

The Implementation phase involve setting up two main components, the CNN and the R-CNN model in the project for further design and testing.

Phase 1 is almost completed, and the completed tasks are listed below:

1. For CNN
	1. Caffe and the python interface at the remote machine is set up
	2. Caffe installation verified with suggested sample (cats and dogs classification)
	3. FER dataset converted Caffe accepted format
	4. BVLC Reference Caffenet modified for 8 classes classification
	5. System 1 training
2. For R-CNN
	1. Python version of Faster-R-CNN at the remote machine is set up
	2. Faster-R-CNN installation verified with the demo provided
	3. Facial component detection dataset created with dlib
	4. R-CNN is modified for facial component detection
	5. Modified R-CNN is trained for facial component detection

The remaining tasks for phase 1 is the merging strategies.

## Phase 2: Experiment phase

Phase 2 involve the design and implementation of merging strategies for images from R-CNN before feed to CNN. Once merging strategies are ready, System 1 will be tested for baseline and System 2 will be tested and compared will different merging strategies. Phase 2 has not been started yet.

# Difficulties encountered

## Limited information from the data set provided.

The dataset used in this project provides some of the emotions provided. Unfortunately, only 327 sequences out of 593 (55%) had been labelled with an emotion category. Since only images with known emotions are usable in training the neural networks, over 4000 images are left unused.

To alleviate this problem, images with similar facial features are grouped the Action Units and Landmarks provided by the dataset. By comparing similar data using Python, 198 more sequences are identified and categorized into an emotion. This increases the amount of training data available for training. Some images can hardly be labelled with any emotions (figure 5.1), and therefore were discarded.



Figure 5.1 Examples of images which cannot be labelled with any emotions.

## Extraction of facial landmarks from the dataset

There is no accessible dataset for facial component detection, which is needed for training the R-CNN model. Instead of hand labeling each of our image with eye, mouth, eyebrow bounding box, we used dlib, a python module to generate the bounding box[11]. The bounding boxes were then exported to PASCAL VOC format annotation. During the first testing of our custom dataset, it is noticeable that the detection rate of eye and eyebrow are much less than other components (~0.001 vs ~0.9). After some testing it is shown that the problem was caused by the bounding box being too tight, which is uncommon when using hand labelled dataset. In the current iteration the bounding boxes are loosen by 10 pixels in both dimension (figure 5.2), which is subject to change after more experiment. The improvement of AP is great as shown in figure 5.3.

 

Figure 5.2 Visualization of boundary of left eye before(left) and after(right) boundary relax



Figure 5.3 AP before and after relaxing boundary, note that some AP before relaxing boundary is too small to visualize

## High error rate of Faster R-CNN model

Another difficulty is although the modified dataset achieved high AP, in single photo testing it often detects both eye as left eye/eyebrow or both eye as right eye/eyebrow (figure 5.4). It is suspected that it is because R-CNN have little success on distinguishing left and right component. As AP is measuring the average rank of the correct bounding box, even if half of the correct bounding box ranked 2, AP will still appear to be high. Current choice of improvement includes treating left and right component as the same or adjusting the input bounding box. Different choices will be tested in the experiment phase.

 

Figure 5.4 Left eye detection by R-CNN with confidence > 0.5

# Future Work Schedule

|  |  |
| --- | --- |
| Jan 15 to Feb 15 | Implementation for merging strategies |
| Feb 15 to Mar 15 | Testing and optimization for the R-CNN + CNN system with different merging strategies |
| Mar 15 to April 1 | Performance analysis |

# Conclusion

Overall the progress of the project is satisfactory as two complex components involving CNN and R-CNN is finished. The team will now focus on implementing merging strategies as they may require tweaking and modifying CNN. Once implementation is completed, analysis on the performance of different techniques will be conducted.

# Reference

[1] N. Sebe et al., Authentic facial expression analysis, Image Vis. Comput. (2006), doi:10.1016/ j.imavis.2005.12.021

[2] Y. Tian, T. Kanade, and J. F. Cohn. Recognizing action units for facial expression analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(2), February 2001

[3] Stewart Bartlett, Marian & Littlewort, Gwen & Frank, Mark & Lainscsek, Claudia & R. Fasel, Ian & Movellan, Javier. (2006). Automatic Recognition of Facial Actions in Spontaneous Expressions. Journal of Multimedia. 1. . 10.4304/jmm.1.6.22-35.

[4] Cohen I., Sebe N., Sun Y., Lew M.S., Huang T.S. (2003) Evaluation of Expression Recognition Techniques. In: Bakker E.M., Lew M.S., Huang T.S., Sebe N., Zhou X.S. (eds) Image and Video Retrieval. CIVR 2003. Lecture Notes in Computer Science, vol 2728. Springer, Berlin, Heidelberg

[5] Kim, BK., Roh, J., Dong, SY. et al. J Multimodal User Interfaces (2016) 10: 173. https://doi.org/10.1007/s12193-015-0209-0

[6] Zhiding Yu and Cha Zhang. 2015. Image based Static Facial Expression Recognition with Multiple Deep Network Learning. In Proceedings of the 2015 ACM on International Conference on Multimodal Interaction (ICMI '15). ACM, New York, NY, USA, 435-442. DOI: http://dx.doi.org/10.1145/2818346.2830595

[7] Mollahosseini, A., Chan, D., & Mahoor, M. H. (2015, November 12). Going Deeper in Facial Expression Recognition using Deep Neural Networks. Retrieved September 27, 2017, from https://arxiv.org/abs/1511.04110

[8] Girshick, Ross and Donahue, Jeff and Darrell, Trevor and Malik, Jitendra. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. Computer Vision and Pattern Recognition

[9] Lucey, Patrick & Cohn, Jeffrey & Kanade, Takeo & Saragih, Jason & Ambadar, Zara & Matthews, Iain. (2010). The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, CVPRW 2010. 94 - 101. 10.1109/CVPRW.2010.5543262.

[10] Jeff Donahue. (2014). BAIR/BVLC CaffeNet Model. Retrieved from https://github.com/BVLC/caffe/tree/master/models/bvlc\_reference\_caffenet

[11] Adrian Rosebrock. (2015). Real-time facial landmark detection with OpenCV, Python, and dlib. Retrieved from https://www.pyimagesearch.com/2017/04/17/real-time-facial-landmark-detection-opencv-python-dlib/