COMP4801 Project Plan

Facial Expression Transfer with Machine Learning

30 September, 2018

Supervisor: Dr. Dirk Schnieders

CHOI Wai Yiu (3035240568)
CHENG Tak Kin (3035240116)
## CONTENTS

1. BACKGROUND .................................................................................................................. 2

2. RELATED WORK ................................................................................................................. 3
   2.1 FACIAL ACTION CODING SYSTEM (FACS) ................................................................. 3
   2.2 GENERATIVE ADVERSARIAL NETWORKS (GANs) ......................................................... 3
   2.3 FACIAL EXPRESSION SYNTHESIS BASED ON FACS .................................................. 3
       2.3.1 2D Action Unit Representation ................................................................................. 3
       2.3.2 3D Action Unit Representation ................................................................................. 3
   2.4 FACIAL ACTION UNIT INTENSITY ESTIMATION .......................................................... 4

3. OBJECTIVE AND SCOPE .................................................................................................... 4

4. METHODOLOGY .................................................................................................................. 4
   STAGE 1: FACIAL EXPRESSION GENERATION ................................................................. 4

   Data Preprocessing .............................................................................................................. 4
   GANimation Model ............................................................................................................. 5
   Generator ............................................................................................................................. 5
   Discriminator ..................................................................................................................... 5
   Model Training and Testing .............................................................................................. 5
   Software Implementation .................................................................................................... 5

   STAGE 2: FACIAL EXPRESSION TRANSFER ...................................................................... 6

   EXTRA STAGE: MOBILE APPLICATION DEVELOPMENT .................................................. 6

5. SCHEDULE .......................................................................................................................... 7

6. REFERENCES ....................................................................................................................... 8
Abstract

This project aims to study the usage of Deep Learning and the Facial Action Coding System (FACS) in facial expression generation and its application in facial expression transfer. In this project, a software is developed using a Generative Adversarial Network model based on Action Units (AUs) annotations to achieve photorealistic facial expression synthesis. In particular, the software is built on top of the GANimation model developed by Pumarola et al. [1] and the software is further combined with the OpenFace toolkit to achieve facial expression transfer.

1. Background

Synthesizing realistic facial expressions from 2D images of human face could be beneficial to a wide range of industries including the film industry and the social media industry. Automatic facial expression generation technique could also help the development of other computer vision research areas such as face recognition and facial expression prediction. Before the success of deep learning, researchers tried to produce new facial expressions by manipulating the facial landmarks with a geometric approach such as triangular geometric deformation [9] and frequency analysis [17], but the results were not satisfactory.

Recently, facial expression generation technology has been substantially improved along with the maturity of deep learning. In 2014, Goodfellow et al. introduced generative adversarial networks (GANs) which are powerful for generative tasks [7]. Since then, many research studies in using various types of GANs for facial expression generation have been published such as CycleGAN [14], IcGAN [5] and StarGAN [18]. StarGAN made a huge breakthrough in facial expression generation. The synthesized facial expression results have become more realistic and natural. However, StarGAN is trained on the Radboud Faces Database (RaFD) [11] which consists of facial images with binary facial expression labels such as happy, sad, angry, fearful, etc. Due to the property of StarGAN, the generation results are limited to the corresponding set of discrete facial expression labels and interpolation between different facial expression is not application [7]. Therefore, StarGAN is not suitable for complex facial expression generation. Although ExprGAN [6] enables the intensity adjustment of a given set of facial expressions, the synthesized results are still limited to a number of discrete facial expressions.

Because of the above limitation mentioned, the Facial Action Coding System (FACS) [12] has started to gain attention in the area of facial expression synthesis. FACS is an anatomical facial expression measurement system which describes facial expressions using Action Units (AUs) which are independent actions of different sets of facial muscle [12]. The combination of deep learning and FACS makes complex facial expression generation possible. Research studies using GAN, autoencoder with FACS has successfully synthesized complex and realistic facial expressions [1][19][20]. Deep learning models conditioned by AU intensity can be applied to achieve facial expression transfer by combining with AU intensity estimation.
2. Related Work

2.1 Facial Action Coding System (FACS)

FACS is an anatomical system which is developed by Ekman and Friesen in for facial expression measurement [12]. In this system, facial expressions are divided into several independent sets of muscle movement. The concept of Action Units (AUs) is introduced to represent the correlation between each independent region of a facial expression and the involved facial muscle. For example, AU1 represent the movement of “Inner Brow Raiser” which involves the frontalis and pars medialis muscle.

2.2 Generative Adversarial Networks (GANs)

In 2014, Goodfellow et al. proposed GANs which are a type of unsupervised deep neural network model structured based on a zero-sum game framework [7]. A classic GAN is composed of a generative network and a discriminative network which are trained simultaneously. The generative network is trained to create realistic sample while the discriminative network is trained to determine whether a sample is produced by the generative network or from a real dataset. GANs are widely used in generative tasks including facial expression synthesis.

2.3 Facial Expression Synthesis based on FACS

2.3.1 2D Action Unit Representation

Zhou and Shi proposed a conditional difference adversarial autoencoder (CDAAE) for photorealistic facial expression synthesis based on FACS [19]. The CDAAE is able to generate a new facial expression based on an unseen input facial image which does not exist in the training dataset while preserving the facial identity. This is achieved by adding a low-level feedforward connection between the encoder and decoder to disambiguate identity changes and facial expression changes [19].

Pumarola et al. introduced a novel GAN scheme conditioned by AU annotation for facial expression generation [1]. The model produces a mapping from a single facial image and an AU intensity vector to a new image of the same facial identity under the desired facial expression. Different from the approach suggested by Zhou and Shi [19], this model achieved an unpaired image to image translation. Instead of pairs of images of the same person under different facial expressions, only images with AU annotations are required for model training, which makes the model more general and flexible. Moreover, Pumarola et al. made the network more robust to background and lighting condition changes by adding an attention layer which limits the network to only manipulate regions of images that are related to producing the new facial expressions [1].

2.3.2 3D Action Unit Representation

Instead of directly generating 2D images, Liu et al. suggested an approach combining 3D Morphable Model (3DMM) [16] with GANs [20]. 3DMM facial expression parameters are first extracted from the input images. The parameters are then used to generate AUs-conditioned 3DMMs. Finally, the output images are rendered from the 3DMMs. Under this approach, the model is able to generate high-resolution facial images.
2.4 Facial Action Unit Intensity Estimation

Baltrušaitis et al. described a method to detect AU occurrence and intensity in real time based on facial geometric features [2]. The problem of individual difference was addressed by using Support Vector Machines (SVM) and Support Vector Regression (SVR) under a person-specific normalization approach based on cross-dataset learning. OpenFace is a popular open source toolkit for facial behavior analysis, including facial action unit detection based on the research of Baltrušaitis et al. It is able to extract 18 kinds of facial action units in 5 discrete levels of intensity [2].

3. Objective and Scope

The main objective of this project is to study photorealistic facial expression synthesis and the usage of Deep Learning and FACS in this research area. The application of facial expression generation in facial expression transfer is also explored in this project. Therefore, the project is separated into two stages. The first stage is facial expression generation. The second stage is facial expression transfer.

In the first stage, we aim to develop a software to generate new facial expression from a single RGB image of human face and AU intensity values. The GANimation model developed by Pumarola et al. [1] will be used which does not require paired training images such that more data is available for training and testing.

In the second stage, the software built in stage one will be further developed to achieve facial expression transfer. There will be two modes of the software: offline mode and real-time mode. In offline mode, the software will receive two images of human face as input and transfer the facial expression from one to another. In real-time mode, the software will take a single static image and a real time video as input. The expressions from the target in the video will be transferred into the face in the static image. If time allows, we will implement a mobile version of the application for a better demonstration.

4. Methodology

This project is divided into three stages: data preprocessing, model implementation, and training and testing. All of the following methodology details are referenced from the approach introduced by Pumarola et al. [14].

Stage 1: Facial Expression Generation

Data Preprocessing

EmotioNet Database contains more than one million facial expression data annotated with AUs, AU intensities and emotion category [4]. We will use 10,000 images from EmotioNet for training and testing.

Another available database is Cohn-Kanade AU-Coded Expression Database. It is a facial expression database prepared by Kanade, Cohn, & Tian [15] and Lucey et al. [13]. It
includes 486 sequences of actions from 97 actors and AUs are marked on peak expression [13][15]. All sequences will be used for training and testing.

As suggested by Pumarola et al., re-annotation of images is required in order to get continuous activation annotations [1]. The re-annotation will be done in the way described by Martinez, Brais, et al. using the OpenFace toolkit [1][2][3].

The AU intensity annotation is formatted as a 17 dimensions vector ranged from 0 to 5. Specifically, the AUs considered in this project are AU01, AU02, AU04, AU05, AU06, AU07, AU09, AU10, AU12, AU14, AU15, AU17, AU20, AU23, AU25, AU26 and AU45.

GANimation Model

We will train and used the GANimation model developed by Pumarola et al. [1] which is published at the Github repository https://github.com/albertpumarola/GANimation. The GANimation model is a GAN model consists of a generator and a discriminator.

Generator

Pumarola et al. introduced an attention-based generator which is a modification of WGAN-GP [1][8]. Instead of directly generating an image, the attention-based generator produces an attention mask and an RGB color mask which are used to generate the final output image [1]. This limits the network to only manipulate the relevant regions of images when synthesizing new expressions. The attention-based generator is used bidirectionally to avoid the need for paired training images [1]. During the learning process, the generator will first generate a new facial image from the input image. Then the new image will be rendered back to the original expression which can be directly compared to the input image.

Discriminator

The discriminator will be trained to evaluate the level of photorealism and expression fulfillment of the generated images. Pumarola et al. proposed a discriminator base on the PatchGan [10] without feature normalization [1]. Moreover, an auxiliary regression head is added on top of the discriminator for AUs activations estimation [1].

Model Training and Testing

For EmotioNet Database, we will use 7,000 image samples for training and 3,000 image samples for testing. For Cohn-Kanade AU-Coded Facial Expression Database, we will use 300 sequences for training and 184 sequences training. The GANimation model will be trained using a single CPU due to hardware limitations. If the training time is too long, we will consider acquiring a GPU to accelerate the training process. At the end of this project, we will analyze and compare our implementation result with the original work by Pumarola et al. [1].

Software Implementation

A desktop software will be built on top of the GANimation model after it is successfully trained. Python will be the programming language for development. This software takes an RGB image of a human face and constantly shows the image under different expressions. The facial expression of the human face in the input image will be changed using the GANimation model. A control panel is provided for adjusting the AU intensity values.
Stage 2: Facial Expression Transfer

In this stage, the functionality of the software will be extended to achieve facial expression transfer. We will embed OpenFace in the software for AU intensity estimation. The software will be able to perform facial expression transfer in both offline mode and real-time mode.

In offline mode, two human face images are taken as the input. First, the software uses OpenFace to extract the AU intensity values from one image. After that, the facial expression of another image is changed based on the AU intensity extracted using the facial expression generation function developed in stage 1.

In real-time mode, the inputs for the software are one static image and a sequence of images sampled from a video recording by a camera in real time. The facial expressions from the sequence of images are transferred to the static image one by one in the same way as the offline mode.

Extra stage: Mobile Application Development

In this stage, a mobile version of the software will be developed. It will be developed for Android platform using Kotlin.

There are two approaches for developing a mobile version of our software which are remote computation and local computation. We will first adopt the remote computation approach in which the computation is done in a server and the mobile application simply sends API requests to the server. Then we will try to embed the trained deep learning models into the mobile application to perform local computation, and see whether there is any improvement in performance.

The advantage of using the remote computation approach is that the software of stage 2 can be reused as the core of the server. The computational cost of the mobile application can be reduced. However, there will be a time lag due to internet communication, and transfer rate is limited by the bandwidth. It will be a significant cost as it needs to upload and download images and even video at the same time.

By performing computation locally, the problem of communication cost can be solved. Nevertheless, there is a trade-off that performance will be restricted by the device. In addition, there is a risk that dependency may not be available on a mobile development environment. Further research and trial are required in this part.
## 5. Schedule

<table>
<thead>
<tr>
<th>Date</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>September 1 – 30, 2018</td>
<td>Basic research on available facial expression synthesis methods</td>
</tr>
<tr>
<td>September 30, 2018</td>
<td><strong>Deliverable of Phase 1</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Detailed project plan</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Project web page</strong></td>
</tr>
<tr>
<td>October 1 – 21, 2018</td>
<td><strong>Stage 1:</strong></td>
</tr>
<tr>
<td></td>
<td>Data Preprocessing</td>
</tr>
<tr>
<td></td>
<td>Training and testing of the GANimation model developed by Pumarola et al. [1]</td>
</tr>
<tr>
<td>October 22 – November 18, 2018</td>
<td>Build a desktop application that can receive AU intensity input and modify an image</td>
</tr>
<tr>
<td>November 19 – November 30, 2018</td>
<td><strong>Stage 2:</strong></td>
</tr>
<tr>
<td></td>
<td>Implementation of chaining result of OpenFace to the GANimation model</td>
</tr>
<tr>
<td>December 1 – December 31, 2018</td>
<td>Deliverable a desktop application using OpenFace to achieve real-time facial transfer</td>
</tr>
<tr>
<td>January 1 – 6, 2019</td>
<td>Preparation for the first presentation</td>
</tr>
<tr>
<td>January 6 – 19, 2019</td>
<td>Working on detailed interim report</td>
</tr>
<tr>
<td></td>
<td>Buffer for overrun</td>
</tr>
<tr>
<td>January 20, 2019</td>
<td><strong>Deliverable of Phase 2</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Detailed interim report</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Preliminary implementation: a desktop application performing facial expression transfer using GANimation model and OpenFace toolkit</strong></td>
</tr>
<tr>
<td>January 21 – February 14, 2019</td>
<td>If time allows, enter the extra stage, otherwise continuous the development and testing of the desktop application.</td>
</tr>
<tr>
<td></td>
<td><strong>Extra Stage:</strong></td>
</tr>
<tr>
<td></td>
<td>API for the mobile application demo</td>
</tr>
<tr>
<td></td>
<td>A mobile application performing remote computation by calling API</td>
</tr>
<tr>
<td>February 15 – April 13, 2019</td>
<td>A mobile application performing local computation</td>
</tr>
<tr>
<td>April 14, 2019</td>
<td><strong>Deliverable of Phase 3</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Finalized tested implementation: finalized desktop application and/or a mobile version of the application</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Final report</strong></td>
</tr>
<tr>
<td>April 15 – 19, 2019</td>
<td>Final presentation</td>
</tr>
<tr>
<td>April 29, 2019</td>
<td>Project exhibition</td>
</tr>
</tbody>
</table>
6. References


