Real Time Avatar Face Transfer With Facial Expression Analysis

1st Hammad Ali
Department of Computer Science,University of the Punjab
Lahore, Pakistan
bcsf17m521@pucit.edu.pk

2nd Muhammad Hasham
Department of Computer Science,University of the Punjab
Lahore, Pakistan
bcsf17m525@pucit.edu.pk

3rdMuhammad Dawood
Department of Computer Science,University of the Punjab
Lahore, Pakistan
bcsf17m519@pucit.edu.pk

4th Daniyal Aftab
Department of Computer Science,University of the Punjab
Lahore, Pakistan
bcsf17m501@pucit.edu.pk

Abstract - There is an ever-increasing demand in the use of internet and more particularly video calling and use of cameras. The modern age bolsters the usage of modern techs which employ various Computer Vision. The importance of techniques to identify the facial attributes have never been more before. This research paper takes into consideration two key techniques for Facial research applications: Face Transfer and Facial Expression Analysis. We will explain basic techniques in this paper to utilize both these domains. The Facial Expression Analysis module would be using a Convolutional Neural Network Model and the Face Transfer module would utilize a 3D model rendering software Blender, which will be used to render python scripts to simulate avatar movement.

I. Introduction

The field of Computer Vision keeps on improving as the time progresses. It would be very useful and handy when such programs will exist that transfers humans face on avatars and can analyze the facial expression of human in camera feed. The Face Transfer can eliminate the need of one having to appear in camera in person, where instead they can use an avatar on their place. The avatar will be capable of steering movements just in the way normal human face would do. The facial expression on other hand can help get insights on the facial expressions of people. It would analyze the face of people and would classify their expressions respectively based on their current moods. This can help analyze the audience and can help get better insights.

This paper consists of two distinct modules with only a slight overlap in between them. These modules are:

1. Face Transfer
2. Facial Expression Analysis

The module of Face Transfer explains how the face of a person can be modelled onto an avatar. It involves explaining the process of how the movements of face will result in the movements of the avatar. The face will be captured by a camera out of which face portion from image will be cropped, discarding rest of the image. Small slices of face will be made i.e. coordinates of various facial areas will be determined. Now as the user will move the face or change facial features, that will result in the displacement of those small slices of face previously made. The coordinates of those small slices will be mapped onto the avatar at every frame of images. This will generate the movement in avatar which will be exactly similar to the movement of face. The output will be the avatar in sync with face movements.

The second module involves Analyzing the facial features and classifying the current expression as either neutral, happy, sad or surprised. For this purpose, a Deep Learning classification model was required. For model to train on, a labelled dataset of images was needed. So the dataset was gathered manually which involved web-scrapping images and people volunteering for their images. After the dataset gathering was completed, the model was built. Model was a Convolutional Neural Network (CNN). The layers and filters of CNN model were configured manually. The Adam optimizer was used. After training for couple hundred epochs, the model started to give an accuracy of over 90% for both training and validation/test sets.

Both modules involve the Face Landmarks technique. This will extract the coordinates of different areas of face. If the face coordinates are obtained, then those coordinates can be cropped discarding rest of the image. The resultant image can be now used further for more pre-processing and modelling steps.

There's a limitation to both of these modules that they depend exactly on how good the captured image is. Meaning how good the quality of captured image is. A good camera and good lighting is very much preferred for both modules to work. If the captured image is of poor quality, the pre-processing step will fail to determine the face which will make face landmarks detection unsuccessful. As both modules rely heavily on face landmarks detection, the program will fail to function if this step fails. Face landmarks will also fail to be determined if lighting of image is low. For this purpose, a good camera and good lighting is paramount for the modules to work.

II. Literature Review

Lots of work has already been done in both of these modules particularly the Facial Expression Analysis. Multitudes of techniques have been used to obtain the Face transfer and Facial Expression Analysis too. Most of these techniques differ at the pre-processing steps. Though some techniques are different altogether.

Face Landmarks have been heavily used in almost all of the techniques. Facial Feature Tracking was used Oliver Schreer [1]. The key principal was to detect the facial features from a specific elevated angel. The displacement in coordinates of facial features resulted in the movement of the avatar. In addition to face transfer, hand transfer was also made possible by detecting the hands. But this would work from a specific position of camera only. The movement of avatar was brought out after employing careful mathematical equations on the facial features' coordinates. Multilinear algebra was also used [2] in order to swap one person's face with the other person keeping the rest of image unchanged. Whole Body Motion Mapping for Avatars also makes use of key features detection [3]. The facial feature extraction is integral for face transfer to work. Andrey Zhmoginov and Mark Sandler [4] explained the inverting of face embedding with convolutional neural network model. They took face construction as a minimization problem. The image gets padded to a size of 256 x 256 and then gets fed to CNN model.

The model to get Face Landmarks is publicly available open source file [5]. This will get the face landmarks from the face(s) present in the image. It returns back 68 different points where each point denotes particular coordinate on the face. The blender can be used to build immersive 3D models [6]. Blender can also natively run python scripts in it. The blender can be used to create various kinds of avatars and then python program can be run on them.

For facial expression analysis, a dataset is needed prior to building CNN model. Only a few portion of the Karolinska Directed Emotional Faces (KDEF) Dataset [7]. KDEF is a set of totally 4900 pictures of human facial expressions. But very little pictures from this dataset are viable for our project. In addition to that, the FERG_DB_256 [8] is another dataset that provides labelled pictures for expression analysis in form of animated characters. The dataset has annotated pictures for these emotions: anger, disgust, fear, joy, neutral, sadness and surprise.

Before predicting the expression, the pre-processing step is of utmost importance. It involves extracting key facial features that result in the formation of facial expressions [9] namely eyes, lips, forehead etc. These are portions of face where serious deformation occurs when a particular emotion is elicited on the face. For instance, the lips change drastically when one is laughing, when one is surprised or when one is sad. Such features impart the key role of making any emotion. Eyes, eyebrows, nose, lips, jaw, chin are involved in almost all kinds of facial expressions [10].

When it comes to which classification serves best for this problem, then most of automatic FERs (Facial Expression Recognitions) use Nearest Neighbors, Support Vector Machines, Artificial Neural Networks and Hidden Markov Model [11]. Under some circumstances, the SVM model outperformed other models [12]. The model built by passing individual facial features into the neural networks showed nice classification results as well. The model wasn't even CNN, a simple feed-forward neural network [13]. The Feature redundancy-reduced convolutional neural network FRR-CNN has also been proposed which involves feeding only particular facial features to the CN model [14]. Y. Li, J. Zeng, S. Shan and X. Chen explained [15] how the FRR-CNN models can perform poorly when the input image is even slightly occluded or damaged.

III. Experimental Work

As our work involves two modules Face Transfer and Facial Expression Analysis, we will be delving into each of these works respectively. Both of these modules overlap at some points particularly the starting phase of both modules is same i.e. capturing face and extracting facial components. We will be delving into both these modules in detail respectively.

A. Face Transfer

Face transfer involves the process of transferring the face of a person onto another person's face or avatar. Essentially, the movements on your face will result in the movement of an avatar. The movement of face is transferred onto an avatar from where it gets its name Face Transfer.

For face transfer to work, it is imperative to detect the face first, essentially the different parts of face such as nose, eyes, mouth etc. The technique called Face Landmarks will be used. This technique involves the process of extracting key facial points which can be later used to distinguish various portions of face.

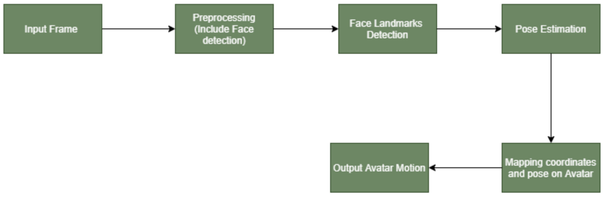
Input Frame
Preprocessing (Include Face detection)
Face Landmarks Detection
Pose Estimation
Mapping coordinates and pose on Avatar
Output Avatar Motion

Fig. 1. Flowchart of proposed methodology for Face Transfer.

In essence, following are key points of how the process of Face Transfer will be accomplished.

1) Capturing the face via camera or video: The basic and most initial step will be to capture the input. This include the process where a camera feed or video feed will be fed to the program. For this purpose, the face of a person will be captured by a camera and then those frames will be sent to program for feature extraction and other purposes. An external camera of good quality will be used to capture the input.

2) Detecting the face portion via the facial landmarks technique: The face landmarks will be determined by using the publicly available open source file called shape_predictor_68_face_landmarks.dat. It will detect the 68 face landmarks points where each distinct point is on the particular area of face.

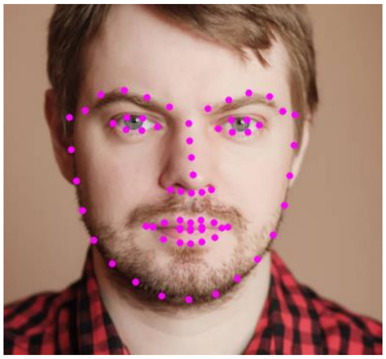

Fig. 2. Sample Face Landmarks detection

3) 3.1.3 Pose Estimation:).: The pose estimation problem described is often referred to as Perspective-n-Point problem [16] or PNP in computer vision jargon. The direct Linear Transformation (DLT) [17] followed by Levenberg-Marquardt optimization [18] will be used to estimate the pose of the image.the SolvePnP method will get be used to determine the pose and will out the rotation and a translation vector. SolvePnP is as follow: solvePnP(InputArray objectPoints, InputArray imagePoints, InputArray cameraMatrix, InputArray distCoeffs, OutputArray rvec, OutputArray tvec, bool useExtrinsicGuess=false, int flags=SOLVEPNP_ITERATIVE).

The rotation can also be represent using Euler angle [19] which will be used to to specify motion in blender.

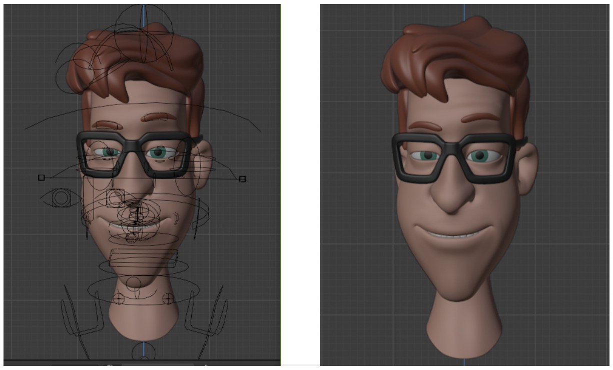

Fig. 3. (a) Blender's Rig Model Demonstration (b) Actual Model without Rig

4) Projecting movement to the Avatar: The euler angle(rotation Vector) for each frame will be mapped to rotation_euler of the headbone of the avatar for x,y and z axis which will cause the movement of the head.

The movement of eyebrow will be calculated with respect to the movement nasal bone and map to the location of the eybrow of the avatar.

The movement of upper lid of the eyes will be calculate by taking avaerage of the two points to get the middle point and subtact it from the lower lid point.

The movement of the mouth will be calculated by subtracting the inner side of the upper lip from the inner side of the lower lip.

B. Facial Expression Analysis

This module involves analyzing the facial expressions. It will classify the face into one of the following four features:
- Neutral
- Happy
- Surprised

For this purposes, a Deep Learning classification model will be built which will utilize a Convolution Neural Network (CNN) architecture. This CNN model will classify the face into one of four features previously mentioned. The model will basically take a pre-processed image and then classify that image into the category which will have the highest accuracy.

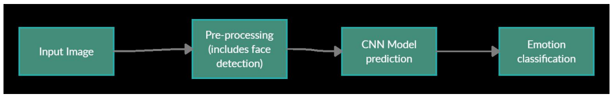

Fig. 4. Flowchart of proposed methodology for Facial Expression Analysis

1) Dataset: Like all classification algorithms need a dataset to train on, our model will also use a dataset which will consist of the labelled pictures of the facial expressions. The dataset was manually gathered as no proper dataset for facial expression analysis was publicly available. So the dataset was manually gathered via web-scraping. A python web-scrapper program was created that visited different open source websites and downloaded images from those websites and saved them on hard drive.

About 5 thousand pictures was gathered initially which passed through different steps of sorting and pre-processing leaving behind approximately a thousand pictures: 250 for each expression.

2) Pre-processing: Before feeding the image into the Neural Network, it will have to be preprocessed first. This preprocessing step involves extracting/cropping only the face portion from the image and neglecting all other parts of image. Only the face area of image will be cropped and image will be overwritten for those face coordinates only. That image will be then converted into grayscale as colored images require more processing. The final image will consist of a grayscale face of the person.

The Face Landmarks technique will be used to this as well. The face landmarks will be detected and only those portions of face will be detected where these face landmarks coordinates are present, ignoring rest of the image. That image will be then converted into grayscale using the opencv's functioncv2.cvtColor(image, cv2.COLOR_BGR2GRAY).

This step is extremely important for the model to work efficiently. If the entire picture is fed into the Neural Network, we will never achieve good accuracy for our model. So the dataset has to be preprocessed before being fed into the model for classification. Likewise, during inference, the image will also be pre-processed before entering the classification model.

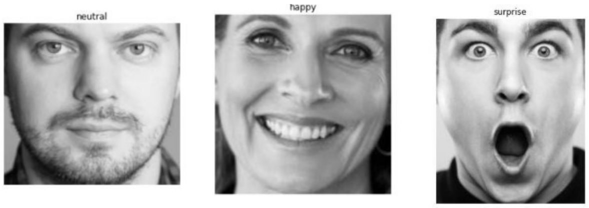

Fig. 5. Sample pictures present in the dataset after being pre-processed

3) CNN Classification Model: Normal Machine Learning classification algorithms wouldn't be able to do this task as it involves careful and meticulous feature learning from training set. So a deep learning algorithm was used. A custom CNN model was designed derived from the LeNet-5 CNN architecture. The layers of this model were tweaked continuously for faster training and faster inference. The final CNN architecture looks as shown in the figure below:

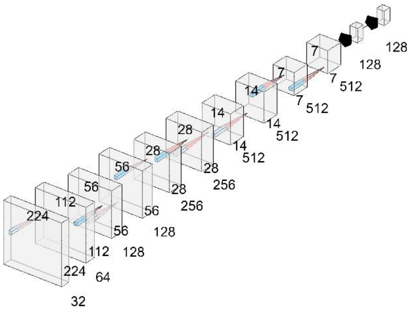

Fig. 6. Architecture of CNN model used

The top and right number being the dimensions of image array and lower number being the Convolution depth or number of filters. The first 10 layers are Conv2D layers and the last two layers are Dense fully connected layers of size 128 each.

The final layer of model has 4 nodes with an activation of Softmax to classify image into one of four expressions. The Adam optimizer was used and the categorical_crossentropy was used as the loss of the model.

4) Training and Inference: The model was then trained on GPU provided by Kaggle platform. The training of model took some time. The validation or test accuracy was extremely poor at start but it started to get better when it was trained for greater number of epochs.

The model was trained for 140 epochs. The following picture explains the training process very clearly.

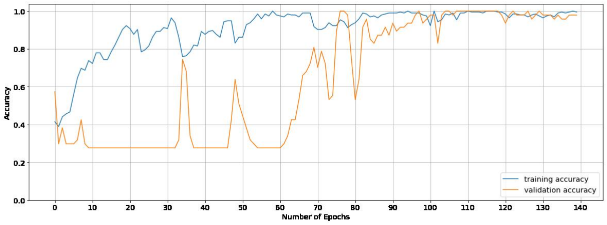

Fig. 7. Graph of Training and Validation Accuracy WRT number of epochs

It should be noted that this model was trained on a limited number of images Dataset which was able to yield more than 90% accuracy.

IV. Conclusion

In this paper we viewed the techniques to explore fundamental methods to incorporate the Facial Expression Analysis and Face Transfer. In particular, a benchmark was set for Face Transfer with Blender, which opens to door to various researches about improvements in this domain. Facial Expression Analysis for 3 emotions was also successfully achieved with high accuracy. As well, the Face Transfer was also successfully accomplished on Blender software.

V. Refrences

[1] Schreer, Oliver, et al. ”Real-time avatar animation steered by live body motion.” International Conference on Image Analysis and Processing. Springer, Berlin, Heidelberg, 2005.

[2] Vlasic, Daniel, et al. ”Face transfer with multilinear models.” ACM SIGGRAPH 2006 Courses. 2006. 24-es.

[3] Spanlang, Bernhard, et al. ”Real time whole body motion mapping for avatars and robots.” Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology. 2013.

[4] Zhmoginov, Andrey, and Mark Sandler. ”Inverting face embeddings with convolutional neural networks.” arXiv preprint arXiv:1606.04189 (2016).

[5] Open source dlib file avaialable at: http://dlib.net/files/shape_predictor_68_face_landmarks.dat.bz2

[6] Takala, Tuukka M., Meeri Mäkäräinen, and Perttu Hämäläinen. ”Immersive 3D modeling with Blender and off-the-shelf hardware.” 2013 IEEE Symposium on 3D User Interfaces (3DUI). IEEE, 2013.

[7] Lundqvist, Daniel, Anders Flykt, and Arne Öhman. ”The Karolinska directed emotional faces (KDEF).” CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet 91.630 (1998): 2-2.

[8] Aneja, Deepali, et al. ”Modeling stylized character expressions via deep learning.” Asian conference on computer vision. Springer, Cham, 2016.

[9] Kumari, Jyoti, R. Rajesh, and K. M. Pooja. ”Facial expression recognition: A survey.” Procedia Computer Science 58 (2015): 486-491.

[10] Chapter 19 ”Facial Expression Recognition” Yingli Tian, Takeo Kanade, and Jeffrey F. Cohn

[11] Căleanu, Cătălin-Danicl. ”Face expression recognition: A brief overview of the last decade.” 2013 IEEE 8th International Symposium on Applied Computational Intelligence and Informatics (SACI). IEEE, 2013.

[12] Michel, Philipp, and Rana El Kaliouby. ”Real time facial expression recognition in video using support vector machines.” Proceedings of the 5th international conference on Multimodal interfaces. 2003.

[13] Saudagare, Pushpaja V., and D. S. Chaudhari. ”Facial expression recognition using neural network–An overview.” International Journal of Soft Computing and Engineering (IJSCE) 2.1 (2012): 224-227.

[14] Xie, Siyue, and Haifeng Hu. ”Facial expression recognition with FRR-CNN.” Electronics Letters 53.4 (2017): 235-237.

[15] Li, Yong, et al. ”Occlusion aware facial expression recognition using CNN with attention mechanism.” IEEE Transactions on Image Processing 28.5 (2018): 2439-2450.

[16] Li, Shiqi, Chi Xu, and Ming Xie. ”A robust O (n) solution to the perspective-n-point problem.” IEEE transactions on pattern analysis and machine intelligence 34.7 (2012): 1444-1450.

[17] Přibyl, Bronislav, Pavel Zemčík, and Martin Čadík. ”Absolute pose estimation from line correspondences using direct linear transformation.” Computer Vision and Image Understanding 161 (2017): 130-144.

[18] Roweis, Sam. ”Levenberg-marquardt optimization.” Notes, University Of Toronto (1996).

[19] Weisstein, Eric W. ”Euler angles.” https://mathworld. wolfram. com/ (2009).