Five Point Feature Recognition of Face and Body for Driver Safety in Real Time Mobile Applications

Eben Sophia.P, Naveen Kumar.S, Selvamani.M, Surya.S, Venkat Balaji.B.S

Abstract: ***In this project step by step implementation of facial recognition and body language recognition using open source algorithm has been explained. The use of body language as a natural interface serves as a motivating force for research in gesture taxonomies, its representations and recognition techniques, software platforms and frameworks which is discussed briefly in this paper. It focuses on the three main phases of body language and face expression i.e. detection, tracking and recognition. Different application which employs body language for efficient interaction has been discussed under core and advanced application domains. This paper also provides an analysis of existing literature related to gesture recognition systems for human computer interaction by categorizing it under different key parameters. Using this application the safety of drivers will be assured.***

Keywords: Facial recognition, android application, driver safety

I. INTRODUCTION

[1]The investigation of the itemized overview introduced in the paper expresses the way that the appearance based hand motion portrayals are more favored than the 3Dbased motion portrayals in the hand motion acknowledgment frameworks. Despite the fact that there are immense measure of data and research productions accessible in both the strategies however because of intricacy of usage the 3Dmodel based portrayals are less favored. The condition of workmanship for uses of the hand signal acknowledgment frameworks present work area applications to be the most actualized application for hand motion acknowledgment frameworks.

Revised Manuscript Received on May 22, 2019.

Eben Sophia.P, Assistant professor, Department of Electronics and Communication Engineering, Karpagam College of Engineering, Coimbatore-641032

Naveen Kumar.S, Student, Department of Electronics and Communication Engineering, Karpagam College of Engineering, Coimbatore-641032

Selvamani.M, Student, Department of Electronics and Communication Engineering, Karpagam College of Engineering, Coimbatore-641032

Surya.S, Student, Department of Electronics and Communication Engineering, Karpagam College of Engineering, Coimbatore-641032

VenkatBalaji.B.S, Student, Department of Electronics and Communication Engineering, Karpagam College of Engineering, Coimbatore-641032

Future research in the field of hand signal acknowledgment frameworks give a chance to the analysts to think of efficient frameworks beating the burdens related with the center advancements in the momentum condition of craftsmanship for empowering innovations motion portrayals and motion acknowledgment frameworks all in all. The mechanical applications additionally require specific progresses in the man to machine and machine to machine communications.

The potential identified with the use of hand signal acknowledgment frameworks in everyday life dependably continues motivating the advances required to understand the solid efficient exact and vigorous motion acknowledgment frameworks.

This paper surveys FFPD techniques, which can be gathered into two noteworthy classes, parametric shape demonstrate based strategies and nonparametric shape display based strategies [2], as indicated by a parametric or nonparametric shape show used in the strategy. Parametric shape demonstrate based techniques are additionally separated into two classes as per a section appearance or all-encompassing appearance show utilized: neighborhood part model based strategies and all-encompassing model-based techniques. Nonparametric shape demonstrate based techniques are additionally isolated into four classifications relying upon their model development process: model based strategies, graphical model-based techniques, fell relapse based techniques and profound learning based methods. In this venture well-ordered usage of facial acknowledgment and non-verbal communication acknowledgment utilizing open source calculation has been clarified. The utilization of non-verbal communication as a characteristic interface fills in as a rousing power for research in signal scientific categorizations, its portrayals and acknowledgment methods, programming stages and casing works which is examined briefly in this paper. It centers on the three principle periods of non-verbal communication and face appearance for example location, following and acknowledgment. Distinctive application which utilizes non-verbal communication for efficient connection has been talked about under center and propelled application spaces.

This paper additionally gives an examination of existing writing identified with motion acknowledgment frameworks for human PC cooperation by arranging it under various key parameters. Utilizing this application the security of drivers will be guaranteed.

In this project well-ordered usage of facial acknowledgment and non-verbal communication acknowledgment utilizing open source calculation has been clarified. The utilization of non-verbal communication as a characteristic interface fills in as a spurring power for research in signal scientific categorizations, its portrayals and acknowledgment systems, programming stages and casing works which is talked about briefly in this paper. It centers on the three fundamental periods of non-verbal communication and face demeanor for example discovery, following and acknowledgment. Distinctive application which utilizes non-verbal communication for efficient connection has been talked about under center and propelled application areas. This paper likewise gives an examination of existing writing identified with motion acknowledgment frameworks for human PC collaboration by classifying it under various key parameters. Utilizing this application the security of drivers will be guaranteed.

II. METHODOLOGY

Implementing the real-time Android Mobile Application using OpenFace algorithms and structures to monitor and alert the drivers from their face expression and body language.

SYSTEM REQUIREMENTS:

OS	WINDOWS 7/8/10
RAM	8 GB
DISK SPACE	5GB
CPU	Intel® Core™ i3-3220 CPU @ 3.30GHz
SYSTEM TYPE	64-bit

OS	MacOS 10 (or Higher)
RAM	8 GB
DISK SPACE	5GB
CPU	Intel® Core™ i5 @ 2.6GHz
SYSTEM TYPE	64-bit

Device Specification

ANDROID VERSION	5.0 (or Higher)
CPU	Qualcomm Snapdragon 625 processor with 2.0 GHz Octa-core CPU and 650 MHz Adreno 506 GPU
RAM	4GB
DISPLAY	1920x1080
CAMERA	Front (5MP), Rear (13MP)
SDK VERSION	21.0

Note: Device configuration should be equal to or higher than the above-mentioned specification. Since this application is real-time image processing app it will consume more RAM and processor time.

SOFTWARE & TOOLS:

ANDROID STUDIO VERSION	2.3.3
GRADLE VERSION	3.3
SDK VERSION	25 (or higher)

The latest versions of development tools can be downloaded from the below link:
https://developer.android.com/studio/index.html

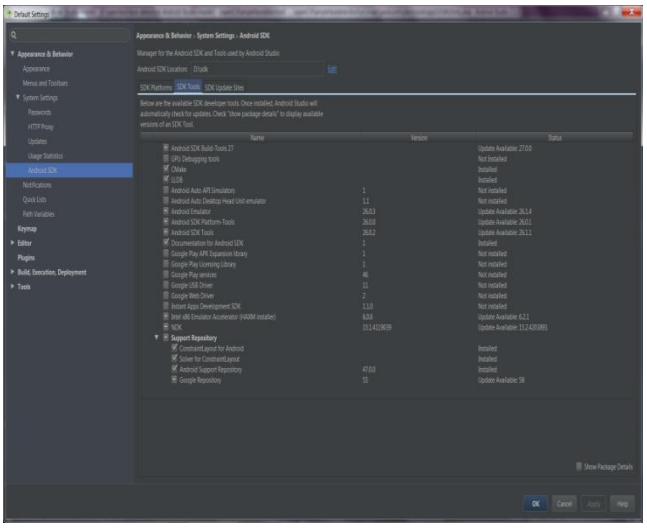

Fig. 1

Application

Splash screen

Fig. 2

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Main screen

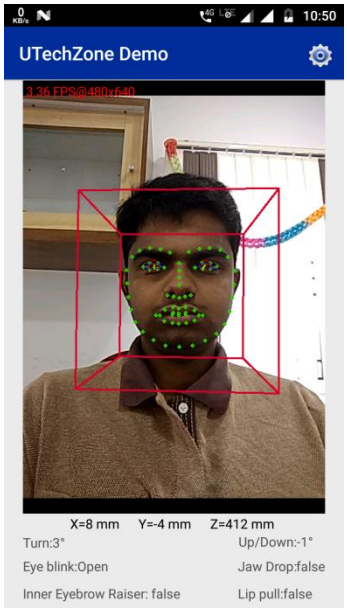

Fig. 3

The main screen of this application will capture the live video from the camera and perform the facial detection based on the settings. Result will be shown in the lower part of the screen.

Settings screen

Fig-a

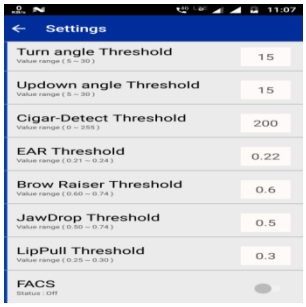

Fig-b

In Camera option, it prompts to select the camera sides.

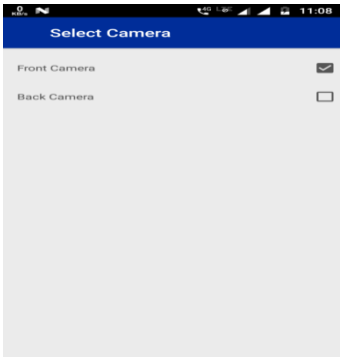

Fig-c

In Resolution option, it prompts to select the resolutions:

Fig-d

- If **Eye Blink** toggle button is enabled, eye blink action unit is enabled and it shows whether the person's eye is open or close. And a toggle button is there for vibrating when the eye blinks.
- If **Pose** toggle button is enabled, it shows the head pose (Rectangular Box) over the face. If it is disabled the Head pose feature.
- If **Gaze** toggle button is enabled, it shows the Gaze lines from the eye. If it is disabled the Gaze feature.
- If **Eye Recognition** toggle button is enabled, it shows the landmark points over the face. If it is disabled the Eye Recognition feature.
- If **Lip Pull** toggle button is enabled, Lip pull action unit is enabled and it shows whether the person's lip is open(True) or close(False).
- If **Auto Focus** toggle button is enabled, it enables the Auto focus feature.
- **SOS** feature is used for Eyebrow Raiser Action Unit. User can give a value in between the range (5-50). If a value is given, it set a frame count value for the Eyebrow Raiser action unit.
- **Turn angle** feature gives the turn angle for the face. User can give a value in between the range (5-30). If a value is given, it set a threshold value for the face angle.
- **Up/Down** angle feature gives the Up &down angle for the face. User can give a value in between the range (5-30). If a value is given, it set a threshold value for the face angle.
- **Cigar Detect** feature gives whether a person is smoking or not. User can give a value in between the range (0-255). If a value is given, it set a threshold value for the smoke detection and it changes the color into blue where the cigarette and smoke is detected
- Action units are calculated in two ways (User Optional).
- Facial Action Coding System (FACS).
- Manual Calculation using the 68 Landmark points.

- If **FACS** toggle button is enabled, the action unit feature use the action units predictors files to calculate the Action Units of the user. If it is disabled, the user has to use the manual calculation. The below features is used to calculate the rest of the Action Units. And it works only if the toggle button is disabled.
- **EAR** feature gives eye blink action unit. User can give a value in between the range (0.21-0.24). If a value is given, it set a threshold value for the eye blink.
- **Eyebrow Raiser** feature gives Eyebrow Raiser action unit. User can give a value in between the range (0.60-0.74). If a value is given, it set a threshold value for Eyebrow Raiser. And if the user is raised his eyebrow for a long time, it checks with the frame count value and shows the signal "SOS".
- **Jaw Drop** feature gives Jaw drop action unit. User can give a value in between the range (0.50-0.74). If a value is given, it set a threshold value for the Jaw drop.
- **Lip Pull** feature gives Lip pull action unit. User can give a value in between the range (0.25-0.30). If a value is given, it set a threshold value for the Lip pull.

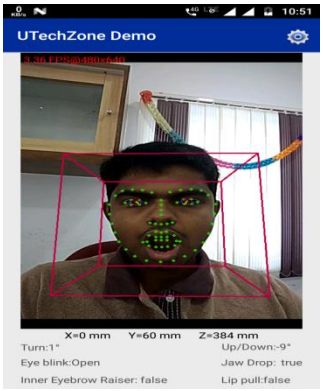

Fig-e

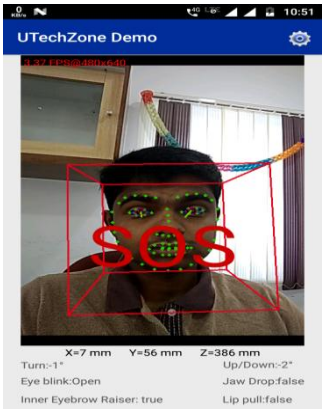

Fig-f

Fig-g

Fig-h

Fig-i

Code Explanation

It has two main modules Java and C++ files, We have achieved combining both JAVA and C++ by use of JNI.
OpenCVSampleDetection directory contains
Java
Jni
JniLibs
Resources

JAVA

1) Settings.java

It contains the main settings menu. We have options for changing the camera sides and the resolutions. And we used Toggle switches to toggle the action units of the person. And the user prompts to enter the threshold values of some actions.

2) CameraChange.java

In this Java class we use camera id's to change the camera to front (99) or back (98). We used shared preferences to save the current state of the camera. We used check boxes for selected camera side but the default camera is front.

3) ResolutionChange.java

In this Java class, we have the options to change the Resolution to 320x240, 640x480, 960x720. We used shared preferences to save the current state of the Resolution. We used check boxes for selected Resolution but the default Resolution is 640x480.

4)SplashScreenActivity.java

It is a splash screen and it contains the App logo.

5) FdActivity.java

onCreate() - App required permission to access the camera. If the app is launched for very first time, it will prompt to the message to grant permission to use camera. Then it will load the settings parameters (based on setting screen or default value if not defined).

BaseLoaderCallback() - A zip file contains all the Action Unit predictors. After the permission process, it will start unzipping. This process will happen at first time when the app is open. Action unit Predictors files are used in LandmarkDetectorWrapper.cpp for Face Recognition and Action Unit process. Action unit Predictors file path is sent to the LandmarkDetectorWrapper.cpp by *InitFunc()* (JNI Function).

fileread() - Action Units like Turn, Eye Blink, Inner Eyebrow Raiser, Up/Down, Jaw Drop, Lip pull and these values are calculated in LandmarkDetectorWrapper.cpp . Value is written in output text file in LandmarkDetectorWrapper.cpp. The function *fileread()* reads output text file and the values are assigned to their corresponding variables.

displayvalue()- It compares with user threshold values with the variables (in *fileread()* function) and it update the Action units variables(Java variables) and the signals like SOS,SLEPPING,LOOK STRAIGHT and it displays in UI in real time.

The processed frames are shown in the camera layout and the corresponding landmark point which is calculated is reflected in the frame also. The frames address is passing to LandmarkDetectorWrapper.cpp by JNI for processing the image. And this happens in every frame.

onCameraframe() - The camera frame is taken and send it to LandmarkDetectorWrapper.cpp by ***RunFaceAR()*** JNI Function and it returns the processed frame.

CPP

1)LandmarkDetectorWrapper.cpp:

It's a main interface between the Java (App UI) and C++ (Openface and OpenCV).

JNI Functions

initFunc() - It assigns the Action Unit Predictors file locations to their corresponding variables in cpp files.

AUPchange() - It assigns the Action Unit settings parameters.

RunFaceAR() - *(Happens in every frame)* :

The frame address is taken and assigned to a Mat variable that contains the frames. The image is converted to grayscale for analyzing the frame. For Detection this function uses one of the two methods 1) HOGG-SVM Detector. or 2) HAAR Detector(default).

This method find the face in the frame and if the frame contains the face, the other Landmark Detector functions will get execute and the head pose points, gaze points & the 68 landmark points for the face is calculated and drawn on the frame. We use ROI in the mouth area to detect whether a person is Smoking by the use of color indicator.

It returns the processed frame.

Action units are calculated in Two ways (User Optional).

1)Facial Action Coding System (FACS)

2)Manual Calculation using the 68 Landmark points.

These Action units' values are written in the output text file.

write_out_landmarks() - This function writes all the Landmarks points, Head pose, Gaze points and Action Units values to the output text file.

III. CONCLUSION

Thus the step by step implementation of facial recognition and body language recognition using open source algorithm has been explained. Using this application the safety of drivers will be assured.

REFERENCES

1. Vision based hand gesture recognition for human computer interaction: a survey by Siddharth S. Rautaray · Anupam Agrawal
2. Facial Feature Point Detection: A Comprehensive Survey by Nannan Wang, Xinbo Gao
3. X. Gao, Y. Su, X. Li, and D. Tao. A review of active appearance models. IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews, 40(2):145–158, 2010
4. L. Gu and T. Kanade. 3D alignment of face in a single image. In Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pages 1305–1312, 2006.
5. Arya DN, Sivanji KL, Reddy R, Sivanantham S, Sivasankaran K. A face detection system implemented on FPGA based on RCT colour segmentation. In Green Engineering and Technologies (IC-GET), 2016 Online International Conference on 2016 Nov 19 (pp. 1-5). IEEE
6. Archana T, Venugopal T, Kumar MP. Multiple face detection in color images. In Signal Processing And Communication Engineering Systems (SPACES), 2015 International Conference on 2015 Jan 2 (pp. 82-85). IEEE.
7. Tripathi S, Sharma V, Sharma S. Face detection using combined skin color detector and template matching method. International Journal of Computer Applications. 2011 Jul 7;26(7):5-8.IEEE
8. Alabbasi HA, Moldoveanu F. Human face detection from images, based on skin color. In System Theory, Control and

Computing (ICSTCC), 2014 18th International Conference 2014 Oct 17 (pp. 532-537). IEEE.

9. Thakur S, Paul S, Mondal A, Das S, Abraham A. Face detection using skin tone segmentation. In Information and Communication Technologies (WICT), 2011 World Congress on 2011 Dec 11 (pp. 53-60). IEEE
10. X. Hou, S. Li, H. Zhang, and Q. Cheng. Direct appearance models. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pages 828–833, 2001.
11. E. Learned-Miller. Data driven image models through continuous joint alignment. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(2):236–250, 2006
12. P. Luo, X. Wang, and X. Tang. Hierarchical face parsing via deep learning. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pages 2480–2487, 2012
13. J Jayalekshmi; Tessy Mathew, Facial expression recognition and emotion classification system for sentiment analysis, 10.1109/NETACT.2017.8076732