Volume Modulation using Hand Gestures

A Special Assignment Report

Submitted in the
Open Elective Course

FUNDAMENTALS OF IMAGE AND VIDEO PROCESSING

By

Name of Students: Ishan Tewari and Labdhi Sheth
Roll No's: 18BCE080 and 18BCE101
B.Tech Sem V

Computer Science and Engineering

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT
INSTITUTE OF TECHNOLOGY, NIRMA UNIVERSITY
AHMEDABAD-382481
NOVEMBER 2020

ABSTRACT

Human imitation for his encompassing surroundings makes him interfere in each details of this great environment. People with hearing impairment are gesturing with one another for delivering a selected message, this technique for correspondence likewise draws in human impersonation regard for cast it on human-computer interaction. The school of vision primarily based gesture recognition to be a natural, powerful, and friendly tool for supporting economical interaction between human and machine. As of late, the interest for various smart home and advanced individual collaborator innovation has expanded drastically.

Over time, analysis in varied fields of computer science have increased the capabilities of sensible homes with refined convolutional machine learning models that constantly analyze sound input for activation phrases and context dependent correction of detected words and phrases in commands. Yet the applications based on gesture recognition stay as a research work.

Google with its MediaPipe is working on its real time had tracking system [1]. Thus using this library we came with a real time application of volume control using hand tracking system which recognizes the palm and its gestures to identify whether to volume up or volume down.

Keywords:
Hand gesture, human computer Interaction (HCI), MediaPipe, hand tracking system, smart home

1 Introduction

1.1 Problem Statement

Communication between people comes from various tangible modes like motion, discourse, facial and body articulations. The primary benefit of utilizing hand motions is to cooperate with computer as a non-contact human computer input methodology. The condition of craft of human computer connection presents the realities that for controlling the computer measures offers of different sorts of hand developments have been utilized .The current exploration exertion characterizes a climate where a number of difficulties have been considered for getting the hand motion acknowledgment strategies in the virtual climate. Being a fascinating piece of the Human computer cooperation hand motion acknowledgment should be hearty for genuine applications, yet complex design of human hand presents a progression of difficulties for being followed and deciphered. Other than the motion intricacies like fluctuation and adaptability of design of hand different difficulties incorporate the shape of signals, continuous application issues, presence of foundation commotion and varieties in brightening conditions. The specifications also involve accuracy of detection and recognition for real life applications [2].

Thus the problem that we are trying to resolve is to use a hand tracking system in smart home devices so as to perform the various functions of these devices using just the hand gesture. Hand tracking system has its own set of barriers as:

1. The dimension of the input
2. Clarity of the video input
3. Presence of multiple hands
4. Skin texture and color of the hands
5. Associating functions to unique hand gestures

Our project surrounds the topic of volume control using hand gestures and thus the pointers to index finger and thumb is used. These tasks are performed using Image and video processing by manipulating the image as input and locating the object of importance. OpenCV library has been used, thus the BGR images are changed to RGB images and to those images we apply MediaPipe algorithm.

1.2 Objective

The current research exertion has an objective of building up an application utilizing vision based hand motions for control of articles in virtual climate. Our application presents a more viable and easy to understand strategies for human computer connection insightfully with the use of hand motions. Elements of mouse like controlling of development of virtual article have been supplanted by hand signals. The intricacy included is with the discovery and acknowledgment stages of the simulated virtual application [3]. The difficulties experienced are loud climate which makes a major impingement on the identification and acknowledgment execution of human hand motions.

Our project explicitly centered on a methodology that included Google's new, quickly developing, and open-source project, MediaPipe [1]. MediaPipe has a few pre-prepared, high exactness picture acknowledgment AI models that sudden spike in demand for a live video feed. There are models for different pieces of the human body.
The image pre-processing uses what is known as feature extraction in order to standardize the input information before we either use it to train a machine learning model or use the pretrained machine learning model to predict something.[8]

Our project has a hand tracking module which tracks down the palm and locates points on the palm called the landmarks. These landmarks are represented by dots and joined by the line which measures the length of the difference between the position of index finger and the thumb to calculate the volume indication as hand gesture by the user.
In the smart home devices, there will be an in build camera which inputs video as frames per second. Our program identifies the hand movements and as soon as the user freezes at a point, that volume is set to the device.

2 Literature Review / Description

Hand signal acknowledgment is a moderately troublesome issue to settle in the field of AI. The majority of these underlying endeavors at making an amazingly precise AI model that identifies hand motions from picture outlines use ordinary convolutional neural organizations. In Sign Language Signal Recognition [7], the task donors took an approach of preparing an AI model utilizing a custom library of 80000 individual numeric signs with more than 500 pictures per sign. Their framework is praiseworthy in showing a refined way to deal with a convolutional neural organization for hand signals. The framework is part into a couple of significant parts, those being a hand discovery framework that utilizes a preparation information base of pre-handled pictures, and afterward a signal acknowledgment framework.

In the paper RGBD Video Based Human Hand Trajectory Following and Gesture Recognition System [9] utilizes a couple extra procedures close by following in two and three dimensional space. To start with, they used two or three strategies to help separate hand highlights from picture outlines, to be specific skin saliency, what's more, movement and profundity based channels. Skin saliency takes the

reality that skin tones are by and large inside explicit reaches to increment the exactness of highlight extraction in picture preprocessing. The movement and profundity channels diminish commotion inside the picture behind the scenes and where tedious highlights and developments emerge.

Data gloves and vision-based recognition are popular and frequently used to capture images for hand gesture recognition [6]. Despite the fact that data gloves have higher exactness, those furnished with numerous sensors are costly. Data gloves [4] are likewise awkward for clients who should wear them for motion acknowledgment. Hence, the vision-based signal acknowledgment framework for a client's bare hand is used in our research. As of late, a mainstream peripheral gadget called Microsoft Kinect [5] for Xbox 360 was produced for hand gesture recognition.

3 Methodology

In this paper, we suggest an approach of using MediaPipe for Hand Tracking and Landmark detection, which is used for modulating the system volume using hand gestures.

3.1. What is MediaPipe?

MediaPipe is a framework which runs on cross platforms(i.e Android, iOS, web, edge devices) for building multimodal (eg. video, audio, any time series data applied ML pipelines. It is currently in development by Google. It contains various cutting edge models like:

1. Face Detection
2. Multi-hand Tracking
3. Hair Segmentation
4. Object Detection and Tracking
5. Objectron: 3D Object Detection and Tracking
6. AutoFlip: Automatic video cropping pipeline

Out of this, we use the 2nd one i.e Multi-hand Tracking.

3.2 Contribution of MediaPipe:

The main contribution of MediaPipe was, [10]

1. An efficient two-stage hand tracking pipeline that can track multiple hands in real-time on mobile devices.
2. A hand pose estimation model that is capable of predicting 2.5D hand pose with only RGB input.
3. An open source hand tracking pipeline as a ready-to- go solution on a variety of platforms, including Android, iOS, Web (Tensorflow.js) and desktop PCs.

3.3 Architecture:

The process of tracking various points in Multi-hand Tracking is basically done by using a pipeline containing 2 different models.

1. *Palm Detection Model:*
   1.1. Operates on a full input image and locates palms via an oriented hand bounding box.

1.2. First, they trained a palm detector as opposed to a hand detector, due to the fact that estimating bounding bins of inflexible items like fingers and fists is considerably easier than detecting arms with articulated fingers.

1.3. The non-max suppression set of rules works properly even for the two-hand self-occlusion cases, like the handshakes. The reason being for the same is fingers are smaller objects.

1.4. Moreover, palms may be modelled with the usage of only square bounding bins [5], ignoring different component ratios, and consequently decreasing the wide variety of anchors through a component of 3~5.

1.5. Second, they used an encoder-decoder feature extractor just like FPN [12] for a bigger scene-context recognition even for small objects.

1.6. Lastly, they decreased the focal loss [13] for the duration of training to aid a massive quantity of anchors on account of the excessive scale variance.

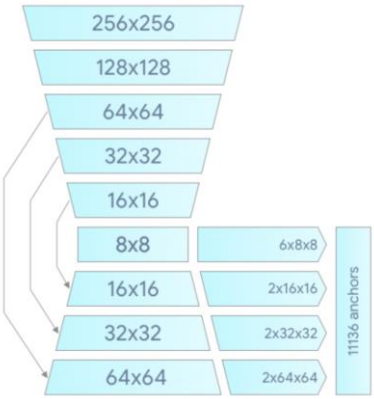


Figure 1:Architecture for Palm detection model

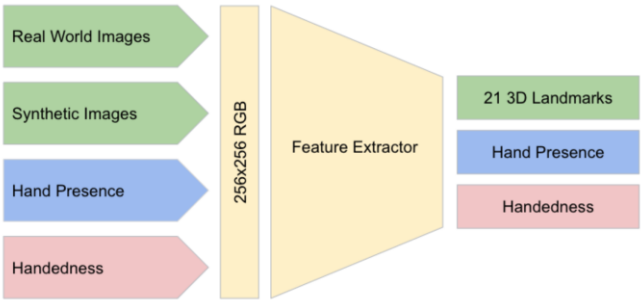


Figure 2:Architecture for Hand Landmark Detection model

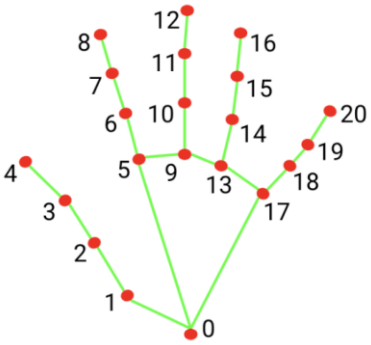
2. *Hand Landmark Detection Model:*

2.1. After running palm detection over the complete image, the hand landmark model performs fairly accurate landmark localization of 21 2.5D coordinates within the detected hand areas through regression.

2.2. The model has three outputs (see Figure 2): [10]

- 21 hand Landmarks which comprises of the x value, y value as well as the relative depth.
- A hand flag indicating the probability of hand presence in the input image.
- A binary classification of handedness, e.g. left or right hand.

2.3. The 21 Landmark points consists of precise location of various points of hands (see figure 3) [11]



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

Figure 3: Landmarks

2.4. The 2D coordinates are learned from both real-world images as well as synthetic datasets. [10]

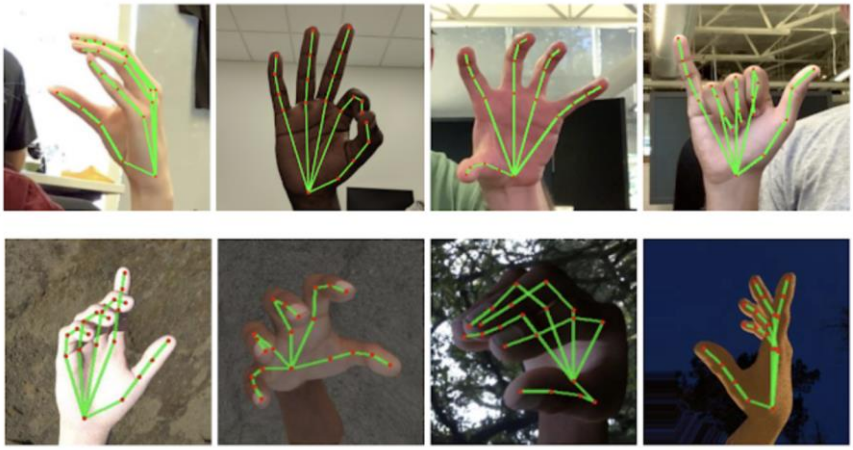


Figure 4: Combined model

3.4 MediaPipe Results:

For the hand landmark model, experiments established that the combination of real-world and synthetic datasets provided the best results.

Dataset	MSE normalized by palm size
Only real-world	16.1%
Only synthetic	25.7%
Combined	13.4%

Figure 5: Results

3.5 Implementation:

So, we used the mediapipe library for hand detection. We used an object oriented programming approach to create a modularized implementation of the hand detector. We made a class *handDetector().* It contains the following methods:

1. *Creating Class handDetector():*
   1.1. def __init__(self, mode=False, maxHands=2, detectionCon=0.5, trackCon=0.5):
      - This class initialized the parameters required for the function mpHands().
      - mpHands is the mediapipe function which we explained above. We will use this function to detect the hands and use the landmark information to modulate voice.

   1.2. def findHands(self, img, draw=True):
      - This function finds the placement of the hand on the screen and draws landmarks over the hand.

   1.3. def findPosition(self, img, handNo=0, draw=True):
      - This function returns id of the landmarks with their pixel location which we will use to manipulate the voice.

2. *Using Class handDetector():*
   2.1. Here we use a library called pycaw to get the details of the volume levels of the computer.
   2.2. Using the module we created, we get the location of the landmark of the thumb and the index finger.
   2.3. We calculate the distance between the two landmarks using the builtin math library.
   2.4. We convert the distance calculated to the range of volume we got using the pycaw library.
   2.5. Finally, we use this metric to set the master volume of the system.

4 Implementation Results

We observed that the volume modulation works pretty well in general scenarios but some of the problems we noticed are as follows:

1. The distance of the hand from the camera is inversely proportional to the accuracy of the change in volume i.e. the further away the hand is from the camera, more difficult it is for the model to predict the change in length accurately.
2. The naive implementation was a bit too sensitive so we reduced the sensitivity with some fine tuning.

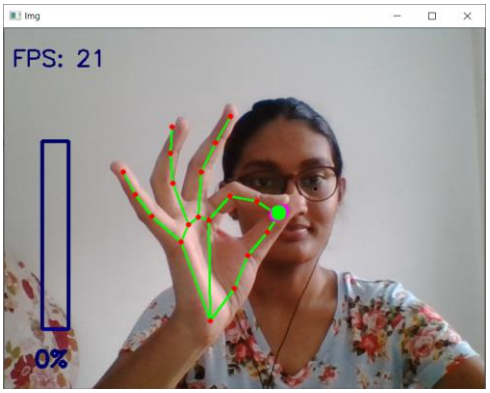


Figure 6: Output at volume 0

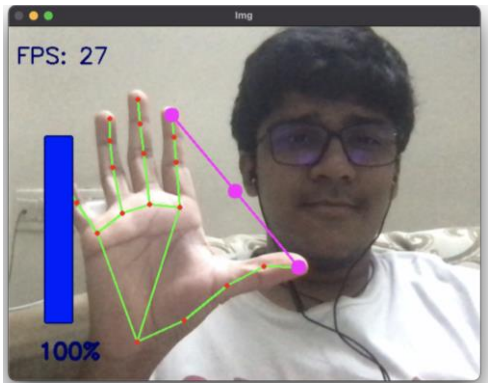


Figure 7: Output at volume 100

5 Conclusion and Future Scope

Our paper suggests a solution where human computer interaction can be merged with daily use smart home devices where constructing an efficient hand tracking system is an important aspect. MediaPipe can be easily used as a tool to accurately determine hand gestures. As per the implementation results we have overcome the problem of always terminating the code and always ending on zero volume. Our future scope would be to come up with some relative distance measure where in the volume perceived remains constant with respect to the hand distance from the camera, expanding the scope of the project by implementing tracking other gestures, tune the model so as to get finer landmarks for more accurate results and ntegrate into smart home devices.

References

[1]. Google AI Blog: On-Device, Real-Time Hand Tracking with MediaPipe (googleblog.com)

[2]. Pang, Y. Y., Ismail, N. A., & Gilbert, P. L. S., (2010), " A Real Time Vision-Based Hand Gesture Interaction", Fourth Asia International Conference on Mathematical Analytical Modelling and Computer Simulation, pp. 237-242.

[3]. Siddharth S. Rautaray1 , Anupam Agrawal2, "REAL TIME HAND GESTURE RECOGNITION SYSTEM FOR DYNAMIC APPLICATIONS", International Journal of UbiComp (IJU), Vol.3, No.1, January 2012.

[4]. S. Jin, Y. Li, G.M.Lu, J. X. Luo, W. D. Chen, and X. X. Zheng: Symp. VR Innovation (IEEE, 2011) 317

[5]. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. CoRR, abs/1512.02325, 2015.

[6]. Hsiang-Yueh Lai,* Hao-Yuan Ke, and Yu-Chun Hsu, "Real-time Hand Gesture Recognition System and Application", Sensors and Materials, Vol. 30, No. 4 (2018) 869–884.

[7]. R. Sharma, R. Khapra, N. Dahiya, "Sign Language Gesture Recognition.," in Sign, June 2020, pp.14-19

[8]. Braden Bagby, David Gray, Riley Hughes, Zachary Langford, and Robert Stonner, "Simplifying Sign Language Detection for Smart Home Devices using Google MediaPipe".

[9]. W. Liu, Y. Fan, Z. Li, Z. Zhang, "Rgbd video based human hand trajectory tracking and gesture recognition system," in Mathematical Problems in Engineering, Jan. 2015

[10]. F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, "Mediapipe hands: On-device real-time hand tracking," arXiv preprint arXiv:2006.10214, 2020.

[11]. https://google.github.io/mediapipe/solutions/hands

[12]. Tsung-Yi Lin, Piotr Dolla´r, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. CoRR, abs/1612.03144, 2016.

[13]. Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dolla´r. Focal loss for dense object detection. CoRR, abs/1708.02002, 2017.

Appendix

Kindly find the code under the github link- https://github.com/Ishan-Tewari/Hand-Gesture_Recognition