# Robotics and Automation - Simulation to ease surgical process

## A DISSERTATION

*Submitted in partial fulfilment of the*

*Requirements for the award of the degree*

*of*

BACHELOR OF TECHNOLOGY

in

MECHANICAL ENGINEERING

**By**

**Vaidehi Som, (2017UME0119)**

**Nishant Kumar, (2017UME0107)**

**DEPARTMENT OF MECHANICAL ENGINEERING**

**INDIAN INSTITUTE OF TECHNOLOGY, JAMMU**

**Jagti, NH-44, Jammu - 181221, J&K, India. (Dec. 2020).**

भारतीय प्रौद्योगिकी संस्थान जम्मू
INDIAN INSTITUTE OF TECHNOLOGY JAMMU
विद्याधनं सर्वधन प्रधानम्

## CANDIDATE'S DECLARATION

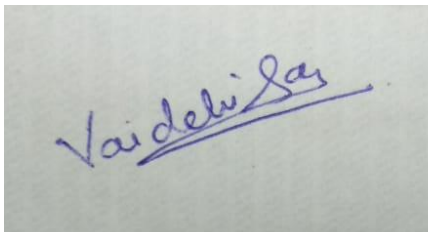
I (We) hereby declare that the work which is presented in this dissertation entitled "**Robotics and Automation- Simulation to ease surgical process**", submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Mechanical Engineering** is an authentic record of work done by my/our own efforts with suitable acknowledgement to all references.

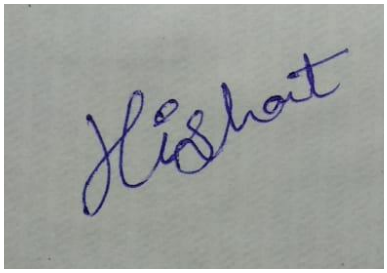
This work has been carried out by me under the supervision of **Dr. Vijay Kumar Pal**, Department of Mechanical Engineering, IIT Jammu during July 2020 to Dec 2020.

I have not submitted the matter embodied in this report for the award of any other degree or diploma to any other institute or university.

Date: 11/12/2020

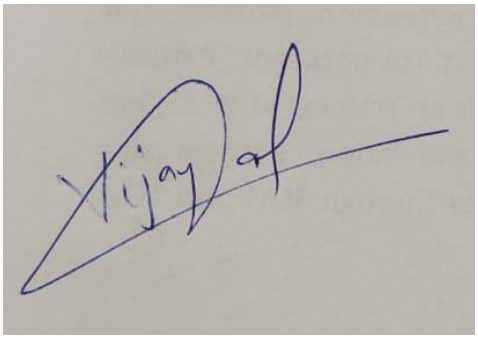
Place: IIT Jammu **Vaidehi Som** **Nishant Kumar**





## CERTIFICATE

This is to certify that the above statement made by the candidate is true, to the best of our knowledge and belief.



Dr. Vijay Kumar Pal, Signature
Mechanical Engineering, IIT Jammu.

# ACKNOWLEDGEMENT

We would like to express our sincere thanks, immense pleasure and gratitude to our supervisors Dr. Vijay Kumar Pal, Department of Mechanical Engineering, IIT Jammu for his constant guidance and support during the project. This project could not have been completed without his supervision. We want to thank Dr. Arvind Rajput, Department of Mechanical Engineering, IIT Jammu for believing in us and providing us an opportunity to work on the topic "Robotics and Automation- Simulation to ease surgical process". At the end, we would like to convey our heartfelt thanks to Dr. Sahil Kalra, Department of Mechanical Engineering, IIT Jammu for their expert supervision.

We are very thankful to my parents & all of our friends for their never ending encouragement in bringing out this dissertation report to the form as it is now.

Date : 11/12/2020

Place: IIT Jammu Vaidehi Som, Nishant Kumar

# ABSTRACT

Hand gesture recognition is one of the most important technologies employed in computer and human interaction.

This plays an even important role in surgical processes, where steps are taken to prevent spreading of infection. For example, doctors have a display in front of them, which aids during surgery by showing the 3d models of body parts, MRI images and so on. The display is used to rotate, zoom in and do other functions to the scans. Similarly, for many electronic equipment, controlling them by gestures is of great interest in the surgical rooms as this reduces the time of surgery.

The complete project includes the following steps:

1. Building of deep learning model to recognize hands
2. Using another model to locate hand landmarks and return the coordinates of all the landmarks
3. Use these detected landmarks to identify different gestures
4. Integrate these gestures with computer to control various interfaces

Two interfaces have been controlled in our project

1. Our computer screen: Gestures were used to control mouse pointer, keyboard and other movements like zoom in/out, swipe, etc
2. Robotic arm: Movement of a simulated robotic arm was controlled using the gestures

The end goal of our project is to make an efficient model that is able to detect these gestures with high accuracy.

# Contents

# 1. INTRODUCTION

Hand gesture recognition systems in operating rooms (ORs) are crucial for browsing and controlling computer-aided devices, which have been developed to decrease the risk of contamination during surgical procedures because doctors have to control the screen during the surgery. Doctors need to use screens to check MRI scans, etc.

Hand gesture recognition is one of the most important technologies employed in computer and human interaction.

This plays an even important role in surgical processes, where steps are taken to prevent spreading of infection. For example, doctors have a display in front of them, which aids during surgery by showing the 3d models of body parts, MRI images and so on. The display is used to rotate, zoom in and do other functions to the scans. Similarly, for many electronic equipment, controlling them by gestures is of great interest in the surgical rooms as this reduces the time of surgery.

However there are many problems in implementing such models. The complexities include having high accuracy and fast response.

Communication with these devices has lots of advantages, and convenience and hygiene are the most important. This may also be employed in robots to hand surgical tools to doctors during surgery.

Gesture communication is more accurate and faster than voice commands. The available work is not of high accuracy and is computationally expensive. We need to work on improving on these two frontiers. Although many gestures can be recognized from a single image frame, we need to build a responsive, accurate system, that can recognize complex gestures with subtle differences between them.

Though we will be focusing on the usage of this model in the surgical room, this can be generalised for use in many more areas, like wheelchair controllers for physically disabled, in

home robotics or in workspace robotics automation. Very little changes will need to be made and the GUI will be ready to use in many other applications.

The ability to perceive hands is very useful and can be used to increase user experience in a lot of domains. Technology to detect hands can be used across various platforms. For example, this technology can be used to detect hand gestures which can be subsequently used to interpret sign language, controlling robots at workplace or at home, and in other similar touchless computer interfaces. The hands detected can also be used to interact with objects in a virtual environment.

While detecting hands is very natural and easy for people, robust detection of the same by computers is a challenging visual task. We present a method to detect gestures using Deep Learning (DL) models. These models are able to detect palms, segment hands from the image frame using the previous detection, and then mark hand landmarks on the hand. Gestures are further recognized using these hand landmarks and are employed in interacting with the computer screen and a simulated robotic arm.

# 2. LITERATURE REVIEW

There are many complex processes involved in the process of gesture recognition. Motion analysis and pattern recognition are some of the major areas where a lot of work is required to be done. Apart from these areas, there are major challenges that are present in the environment when we are trying to make real time identification and prediction of the models. Background illumination and speed of the hand being detected affects the accuracy rate to a large extent. There are many techniques that are being applied to increase the accuracy with which these models are detecting these gestures. All models aim to increase the robustness, scalability, user independence, and real time performance.

Majorly, two kinds of approaches are used to solve the problem of gesture recognition: vision based and sensor based.

### 2.1 Vision based systems

This approach requires the cameras to get the input of images in real time. Models are trained on these images to detect the hands and subsequently cameras.

1. Single camera: Laptop camera, mobile phone cameras
2. Stereo camera: Intel Real Sense camera. Using dual cameras to provide depth information
3. Active techniques: Kinect and leap motion controller

Vision based techniques can be further broken down into two techniques

2.1.1 Model based: In this technique we use skeletal methods to detect hands. Skeletal method is the method that detects the position of hands, it uses hand landmarks to make skeletal like structures on hands. We can then use this skeletal model to define different gestures.

2.1.2 Appearance based: This method uses the dataset that contains the gestures being performed in a video. The model is trained on these videos to learn how a particular gesture is being performed by various people. It learns a gesture by watching many people perform the same action.

### 2.2 Sensor based

These methods use different sensors to gain information about position, velocity, and motion of the hand. Below mentioned sensors can be used for these techniques.

1. Inertial Measurement Unit (IMU): Measures degree of freedom, acceleration, position, etc of hands and fingers to identify gestures.
2. Electromyography (EMG): This measures human muscle's electrical pulses to detect finger movements

# 3. OVERVIEW OF OUR METHOD

We have used vision based method in our project. This is because the sensor method being used is not desirable in many circumstances, like in a surgery room. We do not want an extra sensor which can hinder the surgery. Also, a sensor makes the model dependent, which we do not want.

In vision based systems, we are employing the skeletal based model. The reason is that skeletal models can be easily augmented to include more gestures. Incorporating new gestures in an appearance based model requires training each time. This is not desirable when making the application easy to use and robust.

We have also used a stereo camera. We used the Intel RealSense camera. Usage of the stereo camera has helped us achieve high accuracy as it is a depth camera and provides depth information. This depth information was the reason we were able to segment hands with high precision. This camera also provided us with a large view of the surroundings. Using a single camera would have resulted in a small view which would have reduced the distance width within which the gestures can be identified. This camera can also be used to create a dataset. We did not do so because of time constraints, but using a depth camera helps in the creation of a dataset that can be used in training.

In subsequent sections we discuss our approach and results in detail.

# 4. MODELLING

Modelling of the complete part can be divided into three parts:

1. DL modelling for gesture recognition
2. Computer screen interface control
3. Robotic arm control

All the parts are discussed in detail in their respective columns

## 4.1 DL modelling

It can be further divided into two major steps:

1. Hand tracking- DL model for hand recognition and segmentation, and detecting hand landmarks
2. Declaring hand gestures using the coordinates obtained from hand landmarks

ML pipeline has been used in the first step, whereas the last step has been done using intuitive coding techniques.

### 4.1.1 Hand tracking

Hand tracking solution follows the pipeline

First, a model is run on the complete image and it returns the oriented bounding box around the hand. This box helps with hand tracking. The image is cropped and only the part of the image that contains the palm goes into the further pipeline. This helps in lowering the computational power, as all the resources are concentrated on detecting landmarks and thus the accuracy will be increased.

The model used in palm detection is called BlazePalm. The point to take into account is that we are detecting the palm of the hand, and not the complete hand with finger. Providing the accurate cropped image of palm drastically reduces the need of data augmentation (like rotation, flipping, etc). The network can thus dedicate all its computational power in detecting hand landmarks while being independent of the environment

Blazepalm

The first step in this algorithm is to detect the initial location of hands. This is done by employing one shot detector model. Detecting hands is a complex task because of the way our hands are devoid of notable features unlike the ones present on our face like eyes and lips edges. These contrasting features on the face helps in easy detection. Absence of such features on hands increases the complexity of hand detection. We also have to take in account the complexity of induced due to the fingers. The fingers psoe the problems of occlusion, which further improves the complexity.

To overcome this problem, we train our model on palm detection. Palms are easy to identify as fingers are not present. Also because hands are small, ***non-max suppression model*** was used to find the final bounding box of the palm.


Procedure of Non max Suppression (NMS)

We have the confidence of each bounding box, which is given by s. We have to perform NMS for each box i and find the IOU of that box to the box with maximum confidence. IOU is the ratio of intersection area of both boxes divided by the union of both boxes. To discard the box or to keep it is decided by the following equation

$s_i = \{s_i \quad IOU(M,b_i) < N_t$ or,

$s_i(1-IOU(M,b_i)) \quad IOU(M,b_i) >= N_t \ \}$

where $s_i$ = score of proposal, $b_i$ = box corresponding to proposal, M = box corresponding to maximum confidence, $N_t$ = IOU threshold

The box with the highest s is kept to be the final box

For the final step, cross entropy loss function was used.

Equation for calculating cross entropy loss

This loss can be written as: $-\sum_{c=1}^{M} y_{i,c} \log(p_{i,c})$, where y is the output that the feature is present (y=1) or absent(y=0) as identified by the model and p is the confidence with which the model has made this prediction.

Using all these methods resulted in the accuracy of 86% in the detection of palm.

Detection of hand landmarks

After we have detected the palm, we want to be able to detect the key points on our hands. These key points work as the hand landmarks. We detect 21 total hand landmarks, 4 on each finger and 1 on the palm. These key points give us the 2d location (x, y coordinate) of the landmarks. Regression is used to train the model on the dataset to learn these points.

### 4.1.2 Hand Gesture declaration

A simple intuitive algorithm is used to define the gestures. The state of each finger is identified and the gestures are defined based on the state of each finger. The open and close state of each finger is defined using the x, y coordinates of the landmarks present on that finger. Using mapping of these states, the pre-decided gestures are declared.

For example, we use first finger open, rest all fingers closed gesture for mouse control

Using this straightforward technique has helped in the accuracy of gesture recognition with a small amount of data and training.

## 4.2 Computer Screen Interface control

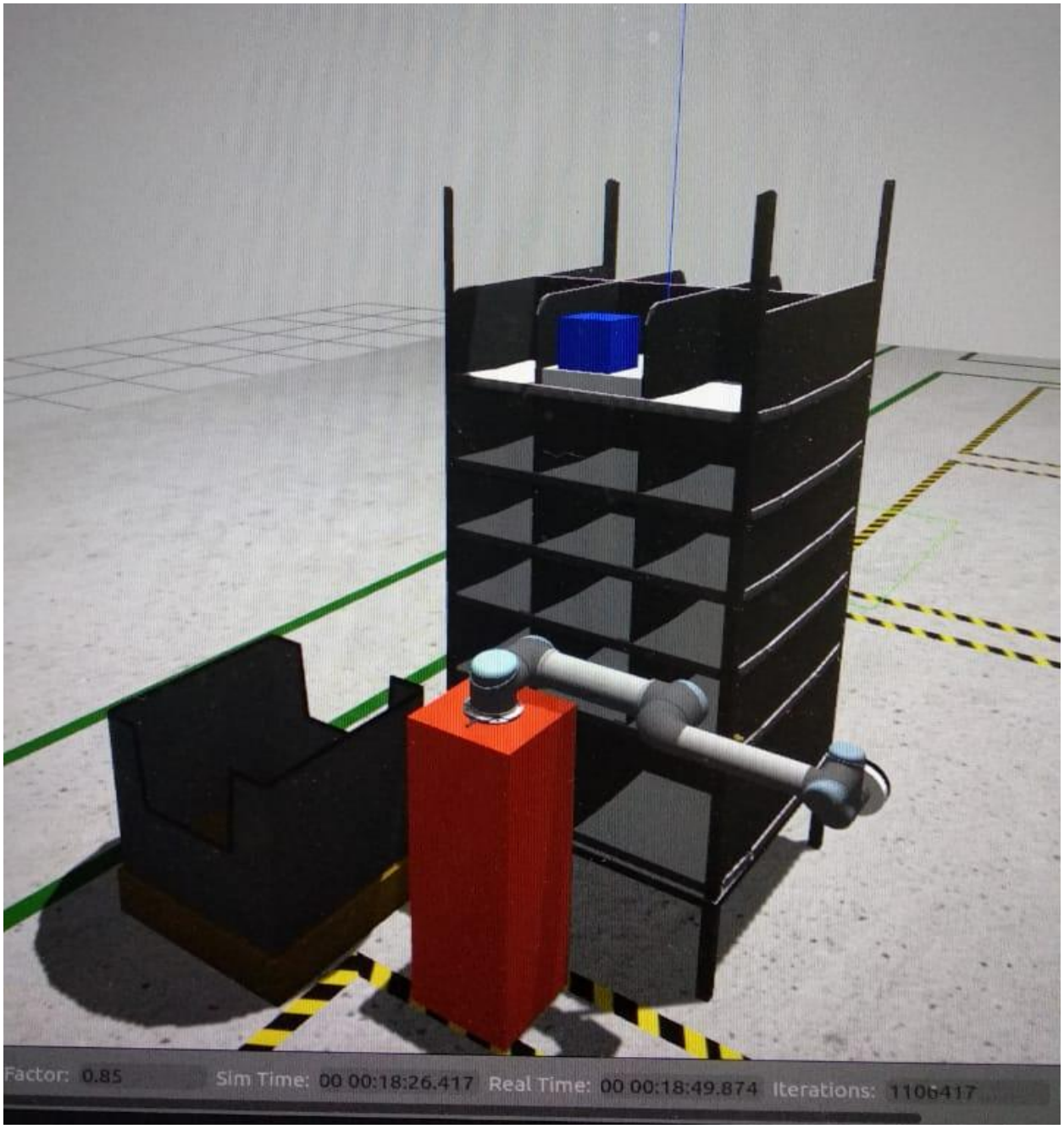
The hand landmarks are returning us the x,y coordinates of those keypoints. These keypoints are mapped to the pixels on the computer screen. The coordinate system used in our model is the same as the conventional computer vision system, where the origin is situated at the top right corner of the screen. Horizontal axis is the x axis increasing in the right direction and vertically downwards is the y axis. The coordinates received from the landmarks are mapped to the computer screen, i.e. the coordinates we are receiving are in the respect of the computer screen origin.

Having the coordinate systems same between the hand landmarks and the counter screen, we are able to control the mouse pointer by moving it to the coordinate of the first finger's upper most

landmark. The computer screen is controlled using the PyAutoGUI library. This library allows us to use the mouse and keyboard functions using codes. When we press the ‘ctrl’ and ‘+’ button together, we are able to perform ‘zoom in’ using the keyboard. We use this technique to press both of these buttons using a particular gesture. Similarly, all the other gestures have been mapped from keyboard/mouse control to the state of fingers.

### 4.3 Robotic arm control

This control involves first the simulation of the robotic arm using ROS, Gazebo and Rviz. The simulated arm is able to pick up objects from a shelf and has the option go towards right or drop it in a nearby box kept on its left.




In our project we have controlled the movement of the robotic arm to perform three functions based on the command it receives via gestures.

End effector points towards right

End effector points towards the box

End effector goes towards the shelf performs it function of picking up the object placed there

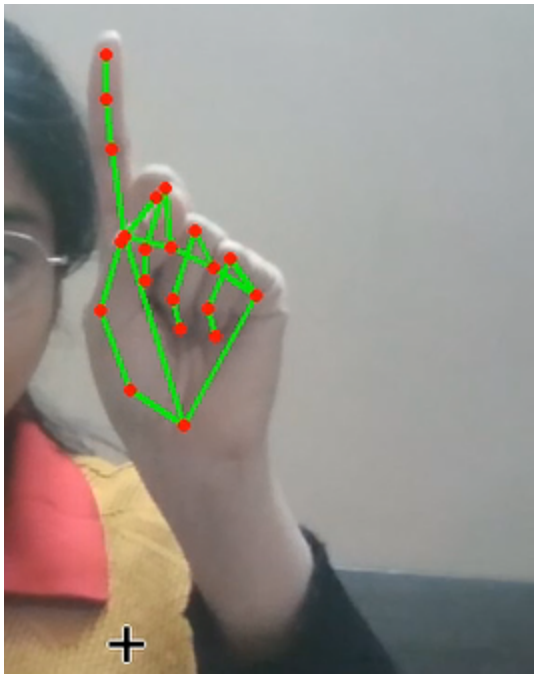
We give the arm three commands and map it to each of the above three mentioned functions.

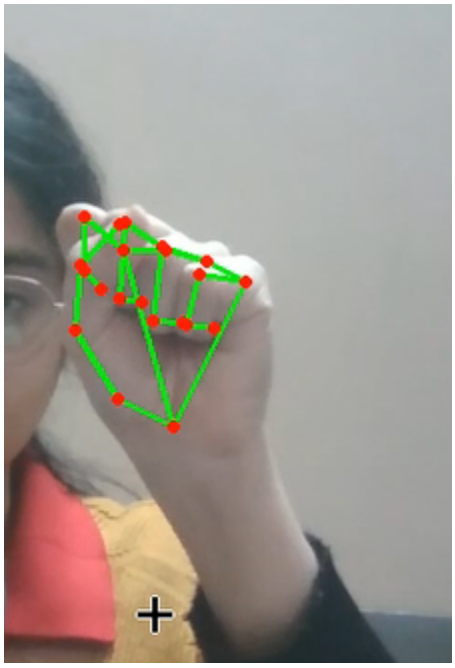
The final position of the end effector at all the three positions are pre-decided. Now, if we will ask the arm to move towards right, it will go to the predefined end effector position. Since there are many links involved, we do not have the complete final position pre-decided. We will have to use ‘inverse kinematics’ to calculate the angle of each joint. The final position is one of the many positions that can be obtained, since there are many angles that will satisfy the condition of base fixed and end effector point given.

Using these methods, we are able to control the three basic operations of the robot.

# 5. RESULTS AND DISCUSSIONS

Results for hand landmarks detection:



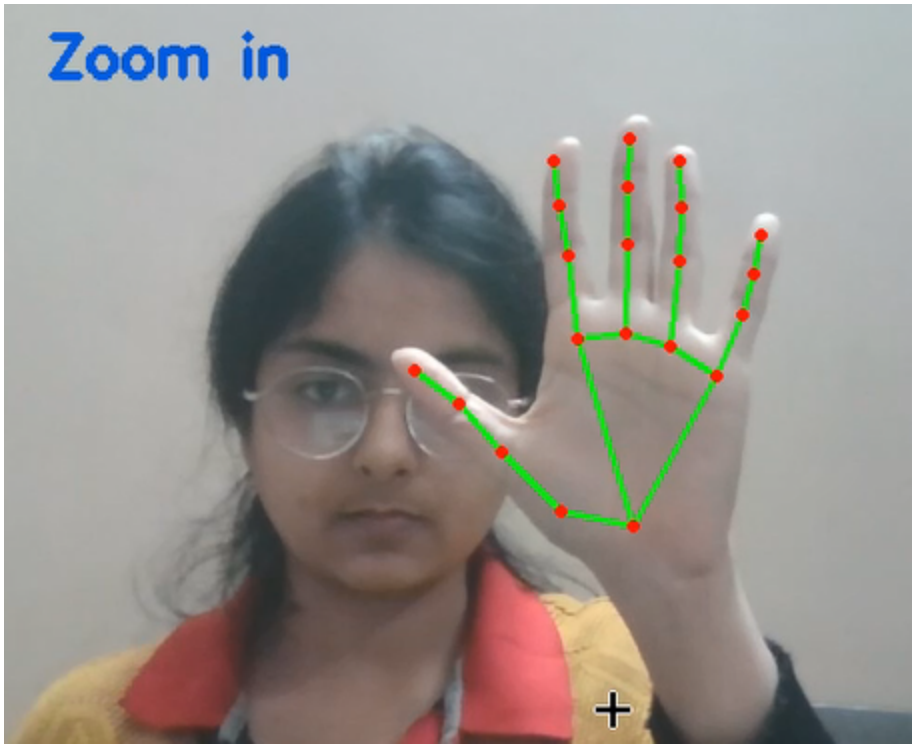


We can see that even in the closed fist, landmarks are being detected properly and easily.

Results for gesture recognition:


Zoom out


Mouse Control




The gestures are also being mapped properly and are being recognized without any problems.

The video control of both the computer screen and the robotic arm were shown in the presentation.

The results obtained are good. All the objectives have been realised, and our model is working with good accuracy.

# 6. CONCLUSIONS

Accurate gesture recognition has several uses on human-robot interaction, user interfacing and virtual reality applications. This is a challenging problem. We, in this project, present a method to detect gestures using palm segmentation and hand landmarks detection.

We have first detected palms using the BlazePalm model. We applied non-max suppression method to get the final bounding box around the palm. Cross entropy loss is used to train the

model. After detecting the palm, the hand segmentation is done to increase accuracy of hand landmark detecting models. After segmenting, we train the model to detect the landmarks.

These landmarks are further used to map to predefined gestures. These gestures are subsequently used to control the computer interface using the PyAutoGUI library. The result obtained are as per the expectations.

Gestures are also used to control the robotic arm's end effector. The arm has been simulated using ROS, Gazebo and Rviz. The arm works properly as defined with no errors.

Thus, our model has been able to work with a good accuracy and provides the expected output results.

# 7. FUTURE SCOPE OF WORK

1. In our results it can be seen that though the hand is being detected, the accuracy is not exceptional. Training of our model has been limited by the computational power available to us. We were not able to train it on a large amount of data, which has affected the accuracy.
2. The model needs to be made more robust.
3. The case when more than one hand is present, how can we control the gesture, is the question that needs to be worked upon.
4. More gestures can be easily included to control more functions of the computer screen, like rotate, swipe, play, and pause.
5. Using PyAutoGUI has led to high sensitivity to changing gestures. The model needs to be made less sensitive to be able to work in a real environment and make the user's experience of usage of the model easy.
6. The robotic arm has only been able to control in the mentioned three ways owing to the computational power constraint. Parallel running of both the hand detection model and the simulation led to the lag in interface which is not desired.

7. The robotic arm can be controlled in more ways, for example, we can make the end effector of the arm follow our finger. The model needs to be augmented to incorporate such features

Above points can be incorporated to make the model usable for a variety of applications. The potential of such systems are huge but we need to make sure that the model displays the accuracy required for a particular application. For example, in a surgical room, the accuracy should be very high is we are controlling a robotic arm. There is noscope of error. Such things needs to be taken into consideration.

# References


[1] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, Kevin Murphy, "Towards Accurate Multi-person Pose Estimation in the Wild", 2017

[2] Valentin Bazarevsky and Yury Kartynnik and Andrey Vakunov and Karthik Raveendran and Matthias Grundmann, "BlazeFace: Sub-millisecond Neural Face Detection on Mobile GPUs", 2019

[3] Francisco Gomez-Donoso, Sergio Orts-Escolano, Miguel Cazorla, "Large-scale Multi view 3D Hand Pose Dataset", 2017

[4] Tsung-Yi Lin and Piotr Dollár and Ross Girshick and Kaiming He and Bharath Hariharan and Serge Belongie, "Feature Pyramid Networks for Object Detection", 2017

[5] Ebrahim Nasr-Esfahani, Nader Karimi, S.M. Reza Soroushmehr, M. Hossein Jafari1, M. Amin Khorsandi1, Shadrokh Samavi, Kayvan Najarian, "Hand Gesture Recognition for Contactless Device - Control in Operating Rooms", 2016

[6] https://www.coursera.org/lecture/convolutional-neural-networks/non-max-suppression-dvrjH

[7] Ming Jin Cheok, Zaid Omar, Mohamed Hisham Jaward, "A review of hand gesture and sign language recognition techniques", 2017