Human Palm Replication with Pose Detection and Control

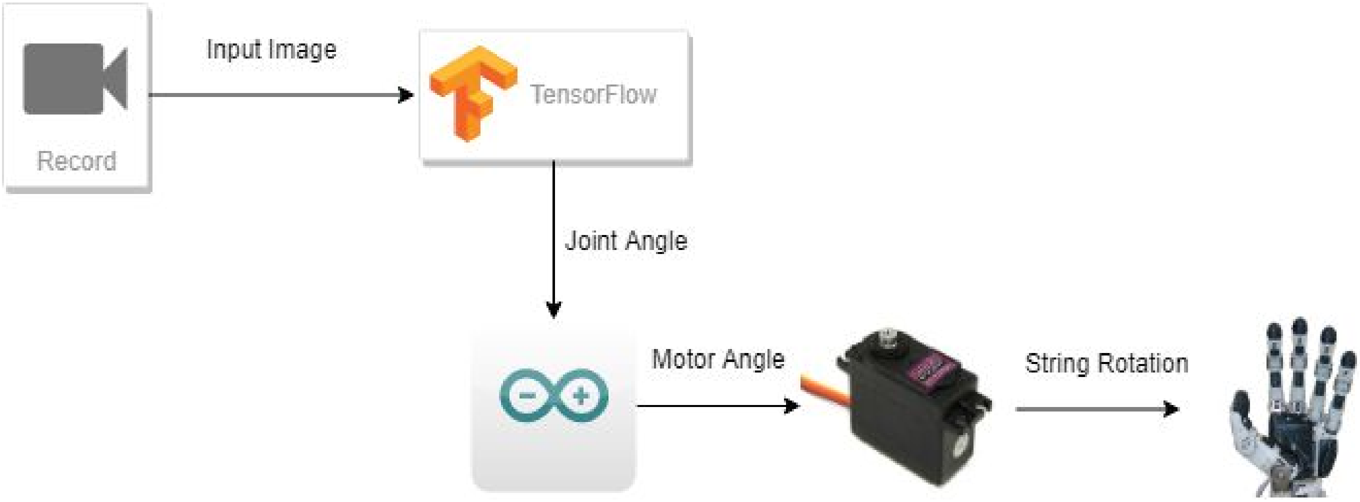
Group No.	7		
Name	Aniket Agrawal	**Roll Number**	160108005
Name	Anuj Gupta	**Roll Number**	160108007
Name	Varshil Gandhi	**Roll Number**	160108016
Name	Vatsal Goel	**Roll Number**	160108018
Name	Mohit Chandak	**Roll Number**	160108026
Name	Shreyas Shubham	**Roll Number**	160108038

Introduction with Functional Block Diagram

The project goal was to design a working model of a human palm and to replicate the palm motion and controls using a video feed of a human palm using cameras.

***Hardware part*:** To work on the 3D printed human palm model, its controls and to design its hardware so as to take processed data as signals from a camera.
Tags: Raspberry Pi / Arduino, Motors, Control mechanisms, Junction Locks

***Software part*:** To work on pose detection algorithms to estimate the pose of the real human palm. (Pose estimation techniques has seldom been applied onto fingers and palms.)
Tags: OpenPose, Computer Vision, Feedback algorithms

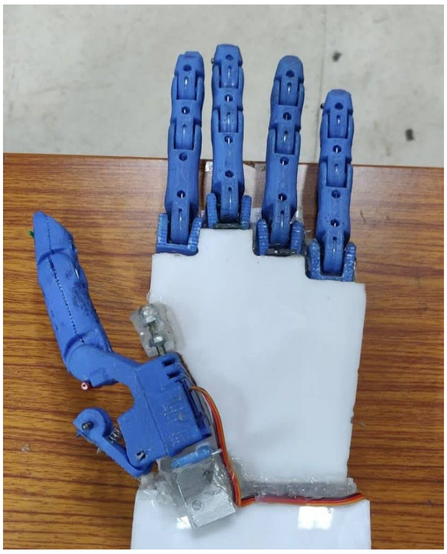


Discussions on Design with Illustrations; Circuit Connections

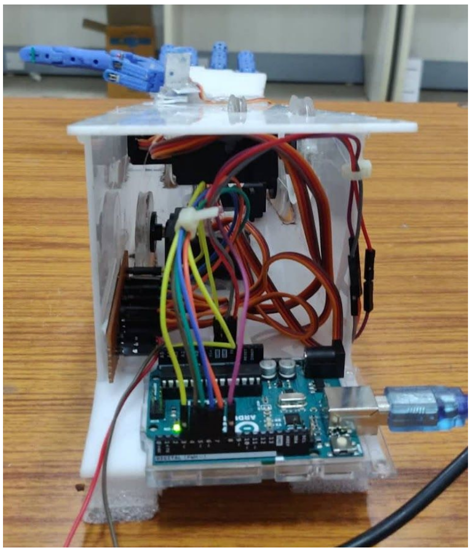
- Hardware Design

a) Palm Design:

The design consists of four fingers and a thumb attached on a palm base. Each finger consists of three smaller components, and the thumb is made of two components, similar to a real human palm. All the fingers have a string making a loop within each finger and connecting all these smaller components, while the other end of the string goes around a motor. Each of the four fingers are connected to a motor each. The thumb is connected to two motors giving it two degrees of freedom. All the motors are controlled by an Arduino.



3D printed Robotic Palm



b) Motor Design:

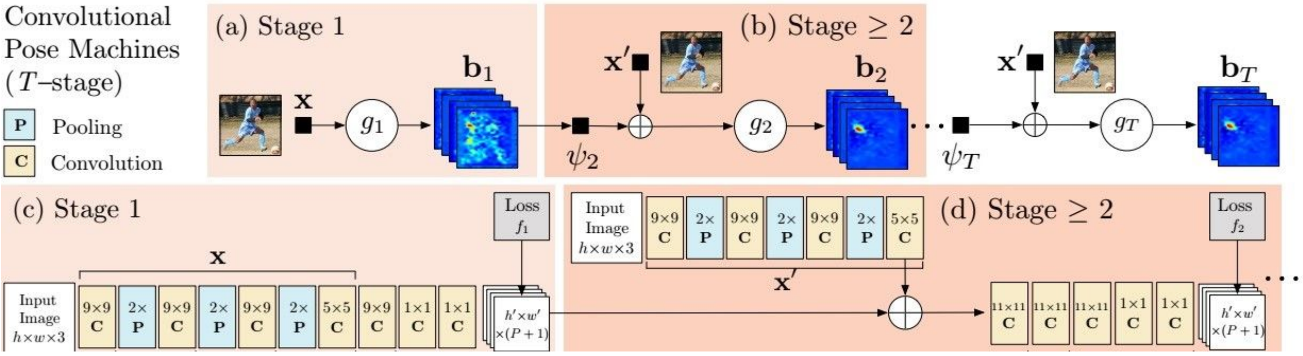
We used **Tower pro MG996** servo motors giving higher accuracy than other servos. Servos were particularly used over DC motors to give finer resolution in angle turned. Since we are using 180 degree servo motors, even upon giving the full angle to the motor, we were unable to rotate the finger completely. So we made wheels with grooves (out of acrylic) and attached them to all the motors, and by wounding the string on this wheel, we are able to achieve the full rotation of each finger at about 130 degrees input to the motor.

Apart from the palm design, we also made a cover (out of acrylic), about the size of a forearm, to keep all the motors and circuitry intact.

- Software Design

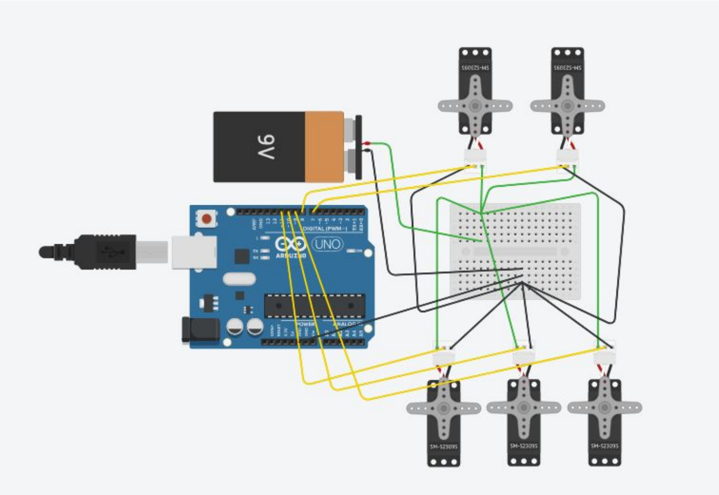
We achieved pose estimation using a three stage convolutional pose machines(CPMs) built on tensorflow-python and kalman filters resulting in higher resolution and accuracy.

a) Architecture: CPM is multi stage model which uses deep convolutional layers to produce belief maps of required joints at each stage. In each next stage, newly calculated features of the image along with the belief maps of previous stage are used as input to achieve a further disambiguating belief maps. The overall proposed multi-stage architecture is fully differentiable and therefore can be trained in an end-to-end fashion using backpropagation. In order to improve accuracy of the architecture, large receptive fields are used in both image and previous belief maps, motivated by goal of achieving long range interactions between different joints of hand. Lastly, such a long multi stage architecture can result in the problem of vanishing gradients. To solve this problem, we calculated loss not only between final output and true belief but sum of loss between each stage output and ideal belief maps replenishing gradients in each stage.



b) Angle calculation: To make hand impervious to different positions in space, we used ratio of lengths of fingers to the distance between finger joints and the lower point of palm. The reference ratios are calculated in the beginning for each user and later angles are calculated observing the new ratios of each finger. Similar procedure was used for the thumb but the reference point was selected along the line of motion.

-Working



Using pose estimation, first we detect all the joints and segments and generate a heat map. Using these positions, we calculate the joint angles with change in position and these angles are transmitted to an arduino via serial transfer. The arduino, maps these angles to its corresponding motor angle. Now, as the motor rotates, the finger bends similar to a human finger in a direction depending upon the direction of rotation of the motor.

Experiments and Results; Performance Analysis

We conducted numerous experiments on which joint positions to take to get our joint angles as precisely as possible. We at first took all the fingertips and the lower point of palm but since the palm detection was largely prone to error, we used the mean of all joints for higher accuracy.

Another important aspect was the tradeoff between accuracy and speed, we took moving average of our angles, and hence we experimented upon the number of values to take, keeping in mind the latency effect.

Lastly, we also ran a number of experiments for the mapping between the input angle to the arduino vs its corresponding output angle to the motor.

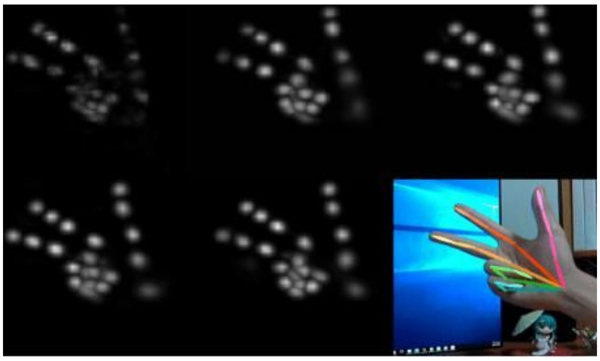


Fig : Example of a multi stage heat map generation

Results :

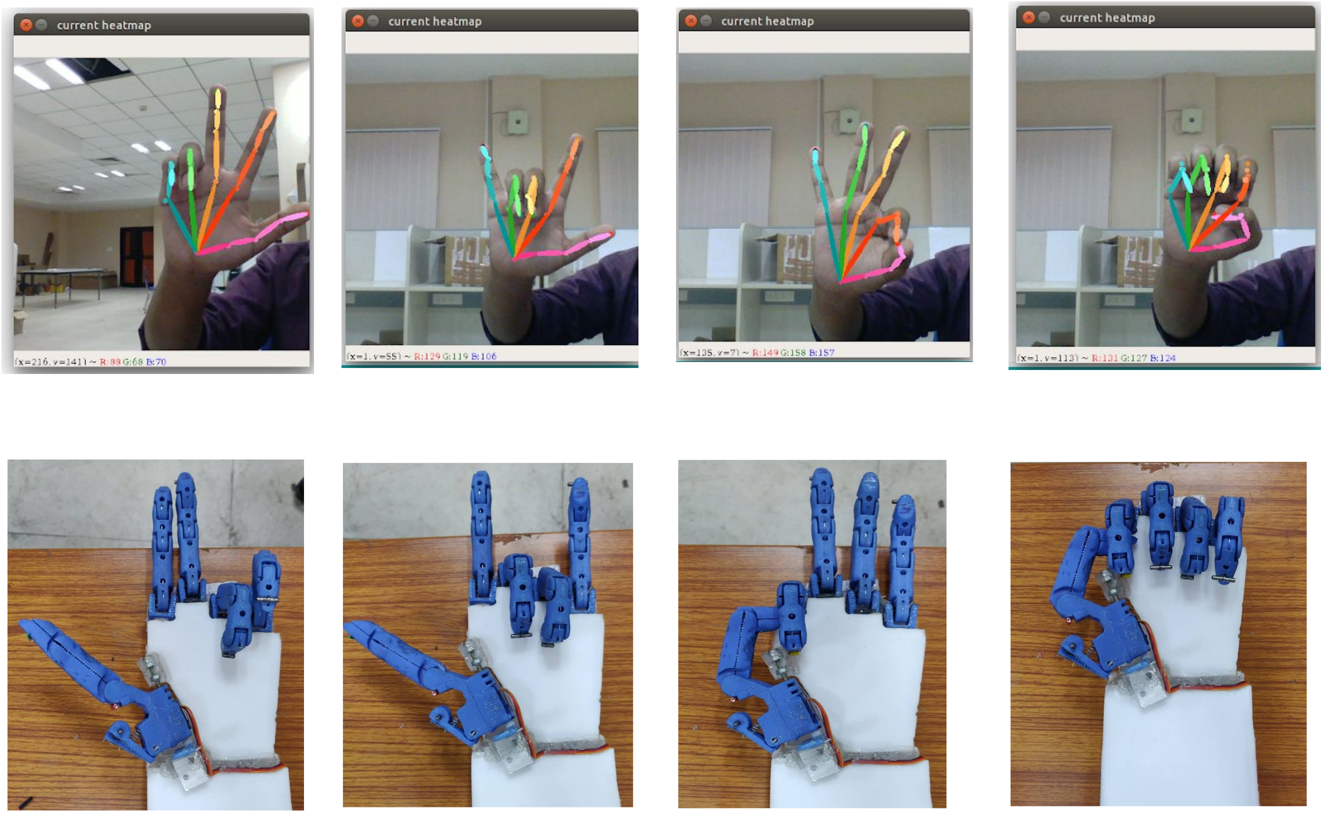

current heatmap
(x=216, y=141) ~ R:88 G:68 B:70
current heatmap
(x=1, y=55) ~ R:129 G:119 B:106
current heatmap
(x=135, y=7) ~ R:149 G:158 B:157
current heatmap
(x=1, y=113) ~ R:131 G:127 B:124

Fig : Real time pose estimation and its corresponding output on the palm

Summary and Future Work

To summarize, we are able to replicate exact motion of our hand in front of a webcam on a robotic palm. We are able to achieve this in real time with a very low latency and good accuracy.

Limitations and Future Work:

***Software*:** Presently, the hand is required to be completely still before the camera and hence it can be inconvenient. Using GANs, pose estimation works with different starting positions of the hand as well. In future, we intend to improvise our model so as to increase flexibility which does not require a still hand and can also analyze poses of the palm even when it is under small motions. In some particular motions, a few joints become a bit too hard to detect which affect the angle accuracy adversely. This aspect can also be looked upon.

***Hardware*:** Even though all our fingers consist of three individual parts, since they are connected by a single string, the last part will turn only after the first two have rotated completely. Similarly the motion of the thumb is also constrained and we fall short in following the various motions a normal human thumb is able to make. Hence, the future work is increasing the degrees of freedom in the fingers and altering the design of the thumb to extend its functionality.

List of Components

#	Item Name	Qty.	Provided by (Dept/Self/Guide)	Price (Rs.)
1	3d Printed Hand Fingers	1	Dept	-
2	Servos	5	Self	1950
3	Mini Servo	1	Dept	-
4	Nylon Rope Thread	1	Self	343
5	Arduino Uno	1	Self	-
6	Jumper Wires	1	Self	-