Hand-Gesture Recognition

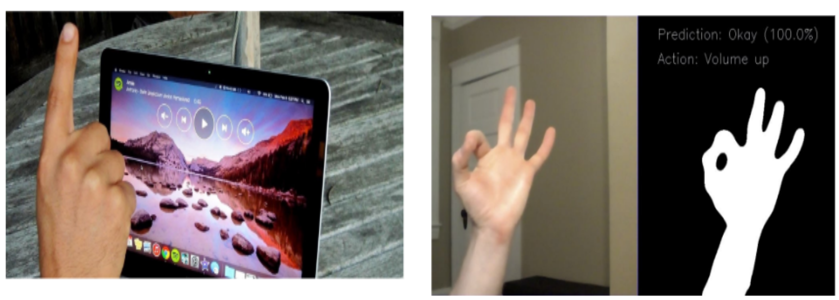
Nitya Mula
nmula3@uic.edu

Ramana Rao Akula
rakula3@uic.edu

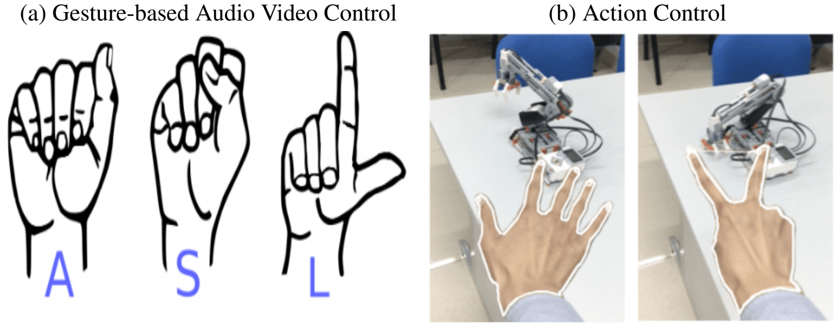
Krishna Garg
kgarg8@uic.edu

1 Introduction

Gesture recognition is the process of understanding and interpreting meaningful movements of the hands, arms, face, or sometimes head. It can help users to control or interact with devices without physically touching them. Gesture recognition has been an interesting challenge in computer vision because of the difficulty involved in segmenting the foreground from the background effectively. It is obvious that there is a semantic gap between how humans look at a picture and how a computer does it. Yet, it has many widely used applications in computer vision and human-computer interaction. Some of these applications include gesture based audio/video control, robotic arm control, sign language system for impaired people as shown in Figure 1. In this project, we tried to develop an application that can recognize the hand gestures from a live video sequence / images.



(a) Gesture-based Audio Video Control (b) Action Control



(c) Sign Language (d) Robotic Arm Control

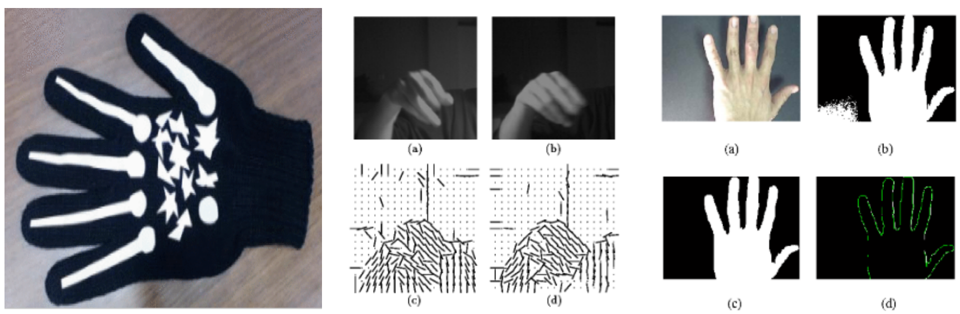
Figure 1: Applications of hand gesture recognition

2 Related work

The problem of hand gesture recognition has been tackled previously by various methods as shown in Figure 2. One of them includes using a special kind of glove called ”the monochrome glove” [2]. This glove has markers made of white tape on it that helps in identifying hand gestures. The glove has markers at the corners of the palm and base of fingers, and white lines through the length of the finger. Based on the bending states and finger angles, the gestures are identified.

The second approach is to use orientation histograms [3]. In this method, from an image input or live sequence of video feature vectors are extracted. It does not use any gesture glove for this purpose. Firstly, the hand area is detected using any edge detection algorithms and separated from the input image or video. The feature vectors representing the hand are extracted using motion estimation. These are later analysed using orientation histogram scheme based on which the gesture is identified. This approach was used to recognize hand gesture for sign language. These vectors are fed to a Hidden Markov Model (HMM) which is a probabilistic graphical model used to predict a sequence of unknown gestures from a set of observed ones.

Another approach uses Fourier descriptors [1] for gesture recognition. It has an edge over other methods mentioned above as it gives features that are invariant to rotation, translation and scaling. They represent the shape of a closed curve at varying form of detail. Also the time taken to process is less as only a less number of points are needed for entire image. Initially hand segmentation is performed using hue, HSV models. Later the Fourier descriptors are extracted and used as features for hand signs. Prediction is done using nearest neighbour or other classification techniques.



(a) Monochrome Glove (b) Orientation Histograms (c) Fourier Descriptors

Figure 2: Existing approaches for hand gesture recognition

3 Problem setup

In this project, we tried to come up with a system that can identify gestures from images/live video sequence. Initially, we made use of contours in the image to identify hand region and then counted the number of fingers in the region using geometric techniques. Later, as we progressed further, we identified some limitations with this approach. This resulted in us shifting to a new approach which made use of neural networks.

The first approach requires the webcam to be working which captures frames and gives the input to the algorithm. In the second approach, a data set of images of various hand gestures is used to train different models and later predict the gestures from the unseen test data set.

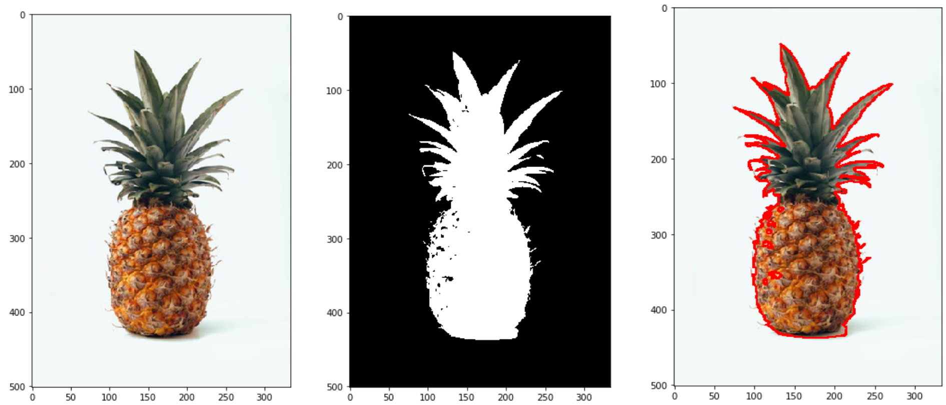
4 Approach

To achieve hand gesture recognition, we initially used the method of finding hand contours and counting the fingers. However, there are some limitations associated with it. This led to shifting to another approach using deep learning.

4.1 Using hand contours

A contour can be defined as a curved line indicating the boundary for same intensity regions in an image. The contour detection can be handled by using cv2.findContours() from opencv library. Figure 3 shows how the contours look on an image.

This approach is followed in two steps namely i) Background subtraction and thresholding ii) Contour extraction. The first step is the most crucial part of this approach because if the hand is seg-



(a) Original image (b) Gray scale image (c) Image with contours

Figure 3: Images showing how the contours are identified and drawn

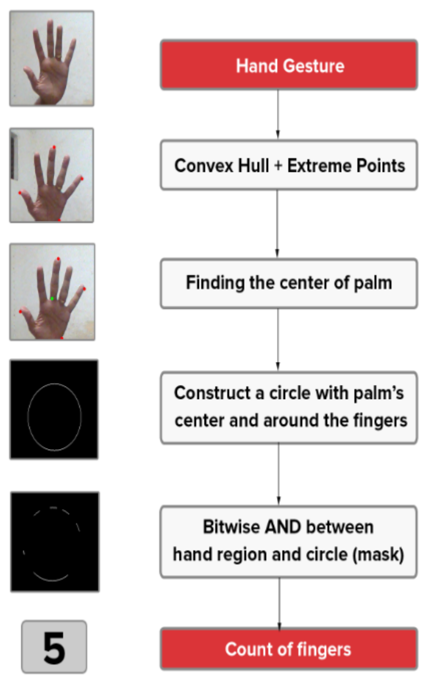


Figure 4: Flow chart for determining finger counts using hand contours

mented incorrectly, he entire algorithm would fail. As a part of the first step, to identify the hand region from the image / video sequence we use background subtraction techniques (frame differencing) and thresholding. The background of the image / video sequence is stored over a certain frame count. Now when a hand comes into the picture, the background image stored previously is subtracted from the current frame to get the foreground image. This way the presence of hand is identified and contours are drawn assuming that the contour with the largest area is our hand.

Figure 4 shows the flow chart for determining finger counts using hand contours. Using contour methods in openCV library, we found out the left and right, top and bottom extreme points. The centroid of the contour is calculated. With this point as the center and the radius as maximum of euclidean distance to extreme points times some threshold, a circle is constructed.Bitwise AND operation was performed between the thresholded hand image and the circular mask obtained above. This gives the number of fingers in the frame by counting the number of transitions from black to white. This way, the finger counts of the hand gesture can be recognized.

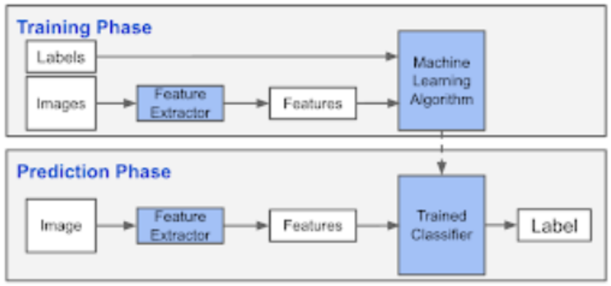


Figure 5: Different phases of using deep learning techniques

However, this approach has a number of limitations. Firstly, this approach can detect up to five finger counts only and thus its scope cannot be expanded. It is too geometric and hand designed. One major drawback is with the assumption that the largest contour (maximum area) in the frame is our hand. If an object larger than the hand is brought inside the frame, then this algorithm fails. However, these drawbacks can be improved by changing the approach to the problem. This is the reason why we choose another approach for hand gesture recognition i.e., by using deep learning techniques.

4.2 Using deep learning models

Deep Learning is a sub-field of machine learning concerned with algorithms inspired by the structure and function of the brain and build using neural networks. A neural network takes in inputs images, processes them in the hidden layers by extracting features and using weights that are adjusted during training. Then the model makes prediction for the unseen data as seen in Figure 5. The weights are adjusted to find patterns in order to make predictions better.

In recent years deep learning has got an upper hand when it comes to computer vision problems. This improvement in the performance of the deep learning models is due to the recent advances in GPU design and architectures. They are well adjusted for training these types of models and are parallel in nature. We have trained models using several neural networks using a hand gestures data set and compared the performances.

Features are extracted from the input image data set and trained on various models both custom and pre-trained models. Predictions are made for unseen test images and the gesture is identified. This approach overcomes the limitations mentioned in the approach using hand contours as the scope of the problem can be increased by spiking the number of gestures. Also, its more sophisticated compared to the hand designed approach previously used.

5 Experimental setup

5.1 Using hand contours

The experimental setup used in this approach is as follows

opencv version - 3.4.2 **imutils version** - 0.4.6 **numpy version** - 1.18.1 **sklearn version** - 0.22.1

The experiment was carried out on a PC with no GPU and it did not give us any problems during the experiment as it was not a GPU intensive application and could be handled fairly by a CPU.

5.2 Using deep learning models

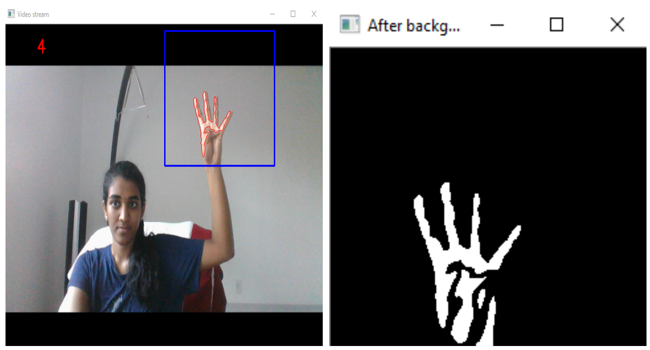
Since the previous approach using contours had many limitations, we decided to use deep learning models for the problem of gesture recognition. We have tried a wide variety of convolutional neural networks such as ResNet50, ResNet152, InceptionV3, VGG16 etc . The experimental setup used here is as follows.

dataset used GPU used keras version

6 Results

6.1 Using hand contours

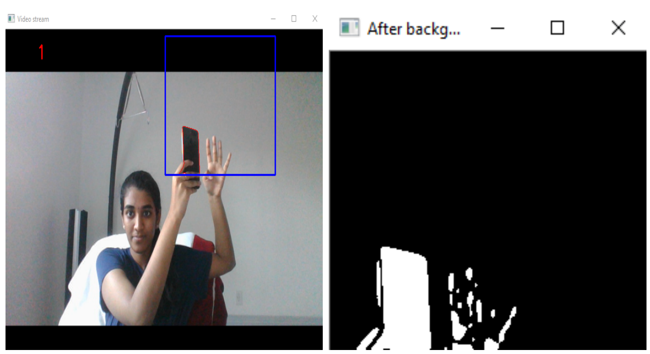
Once the python code is run, it opens up a window showing the live video capture on the webcam. After a particular frame count, the motion within the box is detected and foreground image is separated. Once this is done, the contours are obtained for the hand region as shown in the image below. Now the algorithm tries to get the finger count by counting the change in the transitions from black to white with the help of bit-wise AND operation. This finger count is displayed on the top left corner of the window. As the input gesture keeps changing, the algorithm keeps running and updates the finger count at each moment as seen in Figure 6.



(a) Live video input (b) After background subtraction

Figure 6: Counting the fingers and displaying the count on top left corner of the window

However as previously mentioned when an object bigger than the size of the hand comes into the frame, then the algorithm fails. It can be seen in Figure 7 that when a bigger object comes, its count is predicted to be 1 instead of detecting the hand and giving a count of 4.



(a) Input with interfering object (b) After background subtraction

Figure 7: Wrong finger count due to an object interfering in the input

6.2 Using deep learning models

7 Conclusion

References

[1] Heba M Gamal, HM Abdul-Kader, and Elsayed A Sallam. Hand gesture recognition using fourier descriptors. In *2013 8th International Conference on Computer Engineering & Systems (ICCES)*, pages 274–279. IEEE, 2013.

[2] Hidetoshi Ishiyama and Shuichi Kurabayashi. Monochrome glove: A robust real-time hand gesture recognition method by using a fabric glove with design of structured markers. In *2016 IEEE Virtual Reality (VR)*, pages 187–188. IEEE, 2016.

[3] Hyung-Ji Lee and Jae-Ho Chung. Hand gesture recognition using orientation histogram. In *Proceedings of IEEE. IEEE Region 10 Conference. TENCON 99.'Multimedia Technology for Asia-Pacific Information Infrastructure'(Cat. No. 99CH37030)*, volume 2, pages 1355–1358. IEEE, 1999.

A Contributions

Krishna Garg - Worked on multiple experiments using CNN approach. Tried out various models with different learning rates and batch sizes.

Nitya Mula - Worked on the approach using contours. Identified and put forth the data set to be used for neural networks. Contributed to some experiments using CNN approach.

Ramana Rao Akula - Worked on the approach using contours. Worked on skeleton code for experiments using Keras. Contributed to some experiments using CNN approach.