COMPUTER VISION BASED HOME AUTOMATION

Ankita Pawar[1], Pradnya Shinde[2], Prasad Parsodkar[3], Abhishek Pande[4], Prof. D.G. Kanade[5]
[1,2,3,4]Student, [5]Professor, VIT

Abstract
This paper proposes a smart automation system using computer vision. This system comprises of two fundamental parts, first the camera and second a processing unit for manipulation of appropriate data configured using camera. The concept will be implemented Raspberry pi 3. The system offers the touch free control over the home appliances by enabling the it will simple interaction. Camera for surveillance and vigilance in its range of vision, detect the hand gestures given by user with classifying technique and the conclusions are drawn then signals get derived, followed by the appliances and appropriate actions are executed. It will make the appliances to switch ON or OFF. This will lead to dynamization of control over home appliance and reduction of wastage of power, which is generated by resources going to extinct in near future. The proposed method will use cascade classifier made using haar-like features and convex hull algorithm. System will take an initiative of not only to start an era of less consumption as well as reduction of wastage of electricity but also creates the environment of convenience, which is main objective put forth by this paper.

Index Terms: Haar-like features, Cascade classifier, Hand gesture, computer vision, Convex hull.

I. INTRODUCTION

With an increasing development in modern technology, smart way of living has turn out to be a major part in present era of human life. Automation has brought a revolutionary change in it. Home automation is controlling various home appliances using a computer. Its main aim is to make lives of people simpler and better.

The Raspberry Pi is a single board computer. Simply, the Raspberry Pi system functions like a computer with small setup. It contains GPIO and USB ports. Using these ports and programming the board, we can control the appliances with the sensors as well as interface the camera for surveillance. Raspberry Pi can be used for multiple purposes based on our requirement.

Also, most of the current system require remote control which can be touch based or button based. But in this system whenever there is a need to control an appliances or devices. It can be operated remotely from the place the person is at. So, to overcome this problem the computer vision technology is used.

In this project, computer vision is a great trick of extracting description of surrounding from picture or sequence of picture, so computer vision is very active in research area and rapidly changing. With a fixed single camera mounted in any corner of a room, proposed system can be implemented. Answer to “What about when there is no light, how will the camera detect gestures?” can be by using a thermal camera.

The gestures are recognized used two algorithms, Haar Cascade Classifier and Convex Hull. Haar Cascade Classifier is used to put the camera in recognition mode and then convex hull algorithm is used to read the gestures. The algorithms are briefly explained in this paper.

II. Related Work

Lot of work is going on in the field of Home Automation. There are many systems working on gesture-controlled and voice-controlled Home Automation. Voice controlled systems uses speech processing. Speech processing a quite computationally expensive. Computationally expensive refers to lot of data processing which requires a processor which is computationally sound. A good processor can be costlier. Also, with gesture-controlled systems,

many systems use classifiers to recognize different gestures. Again, the problem with large number of classifiers is the computational power. More the classifiers more is the computational power required. The approach put forward in this paper uses a single Haar cascade classifier and technique called convex hull.

III. Haar-like features

A Haar-like feature takes neighboring rectangular regions at a specific location in a detection window and sums up the pixel in each region and calculates the difference between these sums. This difference is then used to categorize subsections of an image.

The Haar-like feature is calculated as the difference between the dark and bright areas within a kernel.

Each Haar-like feature consists of two or more than two connected “black” and “white” rectangles. Fig. 2 shows the extended set of Haar-like feature. [5]. The value of a Haar-like feature is difference between the sums of the pixel values in the black and white rectangles.

$$f(x) = \sum_{\text{black}} (\text{pixel value}) - \sum_{\text{white}} (\text{pixel value})$$

While training, the input images are resized to 24x24 sized image. The Haar features are calculated all over the 24x24 sized image. At first, the first and the second pixel is covered by the feature and difference is calculated. It moves all over the image. So, the total number of features calculated by single and same sized haar like feature is 23*24. Then the same feature increases the size to 1x4 and goes on like the previous one. Likewise, the total number features calculated by one haar like feature of all sizes is 24*(23+21+19…. +1). Likewise, all types of haar features shown below are calculated on a single image. [3]

It becomes computationally very expensive to calculate these many features on input image. Thus AdaBoost (Adaptive Boosting) algorithm is used which only keeps the features that can really classify positives from negatives. [2-5]

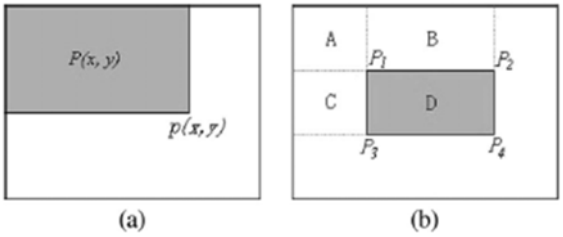


Fig. 1. Integral image

According to the definition of the integral image, the sum of the pixel values within the area D in Fig. 1(b) can be computed by $P1 + P4 - P2 - P3$ where $P1 = A$, $P2 = A + B$, $P3 = A + C$, and $P4 = A + B + C + D$.

Integral image speeds up the feature detection stage of the training. Using integral image, feature calculation becomes much faster and defined.

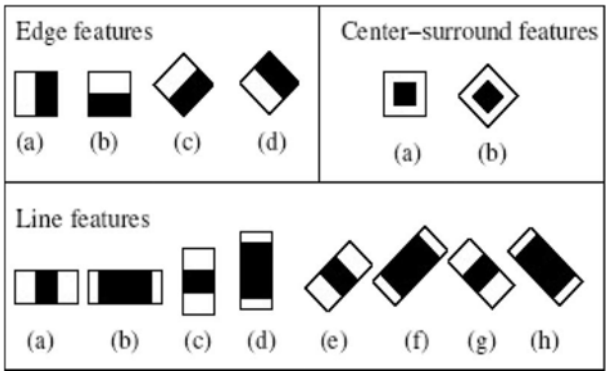


Fig. 2. Set of Haar-like features. [6]

IV. Cascade classifier

A classifier is an algorithm developed to differentiate something and define in which class it belongs. Class is cluster or collection of similar types of objects or data. While training a classifier, two classes, i.e. positives and negatives are provided. Positive is some data to be classified or detected and negative is some data not to be classified or detected. Now, in training, the classifier takes out some features that can really differentiate the positives from the negatives. [1]. The work of the classifier is to correctly tell, to which class some data belongs. Cascade is series of something and cascade classifier is series of classifiers. Cascade Classifier has many number of user defined classifier. The features extracted while training are distributed in these classifiers. As not all the features that can differentiate data are present in a single classifier, each classifier in the cascade classifier is known as weak classifier.

$$f(x) = \sum_{t=1}^{T} \alpha_t . h_t(x)$$

These weak classifiers are cascaded to form a single strong classifier. $f(x)$ is a strong classifier that is a weighted sum of weak classifiers. $h_t(x)$ is a weak classifier, α_t is the weight of the classifier. Weight of a weak classifier in a strong classifier is how much importance is given to that weak classifier or the best weak classifier among all is given the greatest weight. [7-8]

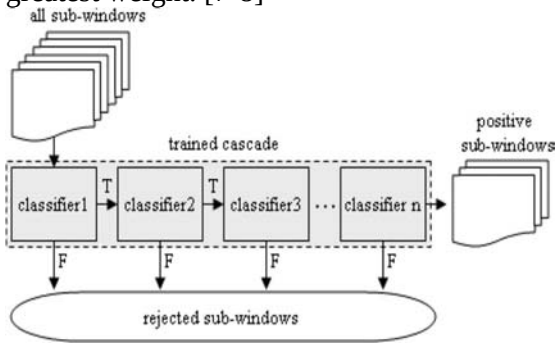


Fig. 3. Detection of positive sub-windows using the trained cascade.

When a test frame is given to a classifier, the kernel moves over the image and each classifier tries to find the features its contain, if a weak classifier finds the same features present in the test frame as in the classifier, the test frame is declared as positive and given to the next weak classifier to do the same. If a weak classifier gives negative result to a frame, the frame is directly discarded from the whole classifier and is not passed forward to further feature computation. This saves a lot of computational power of the processor. This is the main advantage of cascade classifier. For a weak classifier to work well it should have high hit rate and low false alarm rate. Hit Rate is the percentage of positives correctly classified or labelled as positives. False alarm rate is percentage of negatives falsely classified or labelled as positives. [10]

V. Convex Hull

The convex hull of a finite set of a set P of n points in same plane is an important geometric concept. It is defined as the smallest convex polygon for which each point in P is either on the boundary of the convex polygon or in its interior.

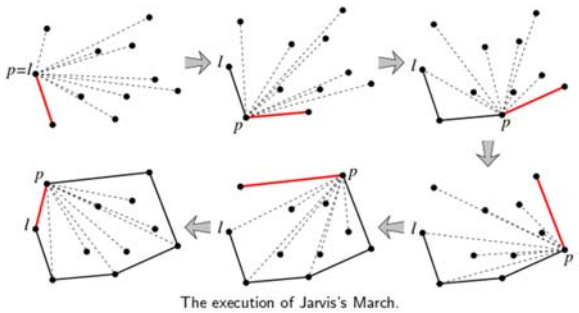


Fig. 4. Execution of Jarvis's March.

The concept of Jarvis March is simple, we start from the leftmost point (or point with minimum abscissa value) and we keep wrapping points in counterclockwise direction. The points on the output contour of Jarvis march are called as hull points. [4]

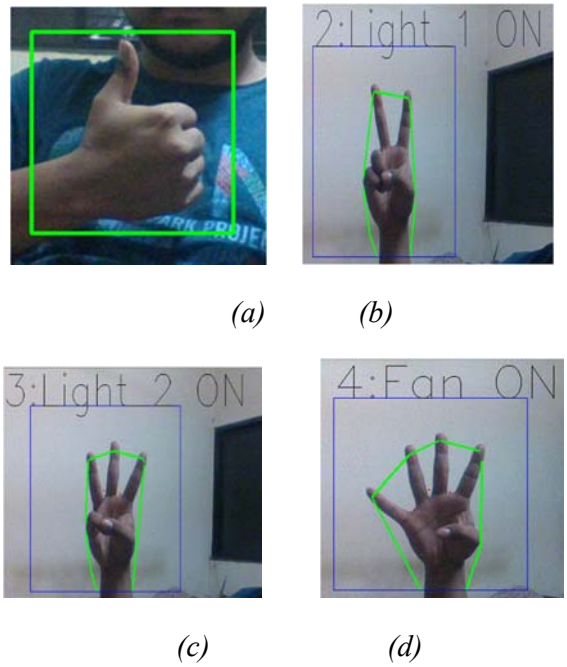
Following is the step by step algorithm. [4]

a) **1)** Start with p as the leftmost point. **2)** Do following while we don't come back to the start point(p) (or leftmost) point. Draw a line between the start point p and the closest rightmost point(q).
b) Now, q is the next point, any random point(r) is connected for the instance to q. Cross product of all remaining points is taken with q(except p, q and r). If the answer is negative, then the point is in the right of the q-r line. If the answer is positive, then the point is on the left of the q-r line.
c) If all the answers are positive, it means all the points are on the left of the point r and thus r is our hull point.
d) If answer to some points is negative, it means there are some point in the right. Thus, the temporary q-r is removed, and q is connected to the right most point, and the same steps are repeated until we reach our start point s.

VI. Results

The recognition system not only just depends on the sample set mention in Fig.1, but also on the number of stages. In the training stage, stage number is an important factor as well as sample

set, however, it does not mean that the bigger the stage number, better is the result [9].
The proposed system was implemented on Raspberry Pi 3 Model B using a USB camera.



(a) *(b)*

(c) *(d)*

Fig. 5. Some of the gestures used and tested

VII. Result Analysis

As two techniques are used in this system, analysis of both techniques is carried out separately.

- Haar Cascade Classifier

Creating a Haar classifier can be tricky at times. The efficiency of the classifier is majorly dependent on Hit rate and False alarm rate of the classifier.High Hit rate and low false alarm rate is desired. The time complexiety of an Adaboost classifier(haar classifier uses adaboost) is $O(nd^2)$. Where n is the number of training samples and d is the number of weak classifiers.

Hit Rate	0.9279
False Alarm Rate	0.5673

- Convex Hull

Convex hull being a basic mathematical concept has very high effiency and comparatively less computational cost. The time complexiety of the convex hull algorithm(Jarvis-march) is O(nh), where n is the number of points in the set and h is the number of points in the hull.

Hit Rate	0.97
False Alarm Rate	0.20

VIII. Conclusion

This proposed approach uses two recognition techniques i.e. Haar Cascade Classification and Convex Hull algorithm. As stated earlier, classifiers used in large number slows the processor using most of the computational power of the processor. In the system proposed, only one cascade classifier is used. It has time complexity of $O(nd^2)$, to increase accuracy,n is to be increased but it has effects on the time complexity. Thus using a single haar cascade classifier takes less computational power of the processor. Haar classifier being more robust is used to put the system in recognition mode.

After the system is in recognition mode, convex hull is used to recognize gestures. It has a complexity of O(nh). It is a simple concept used in geometry which finds convexity defects. It takes small amount of processing power. This is a great benefit of this system over other systems

REFERENCES

[1] Y. Amit, D. Geman, and K. Wilder. Joint induction of shape features and tree classifiers, 1997.

[2] Qing Chen, Nicolas D. Georganas, and Emil M. Petriu, “Hand Gesture Recognition Using Haar-Like Features and a Stochastic Context-Free Grammar”

[3] Paul Voila and Michael Jones, “Robust Real-Time Object Detection” in *International Journal of Computer Vision.*, 2001,pp. 289–296.

[4] “Convex Hull (Jarvis’s Algorithm or Wrapping)” [online]. Available:
http://www.geeksforgeeks.org/convex-hull-set-1-jarviss-algorithm-or-wrapping/

[5] R. Lienhart and J. Maydt, “An extended set of Haar-like features for rapid object detection,” in *Proc. IEEE Int. Conf. Image Process.*, 2002, vol. 1, pp. 900–903.

[6] “Defining an (initial) set of Haar Like Features” [online]. Available: http://stackoverflow.com/questions/27322848/defining-an-initial-set-of-haar-like-features

[7] T. M. Inc., "Train a Cascade Object Detector," [Online]. Available: http://www.mathworks.se/help/vision/ug/train-a-cascade-object-detector.html#btugex8. [Accessed Nov 2014].

[8] Sander Soo, "Object detection using Haar-cascade Classifier", Institute of Computer Science, University of Tartu

[9] Dakuan CUI and Andre L. C. Barczak, "Hand Detection and Gesture Recognition using ASL Gestures"

[10] Eric Emer, "Boosting Algorithm",[Online]. Available: http://math.mit.edu/~rothvoss/18.304.3PM/Presentations/1-Eric-Boosting304FinalRpdf.pdf