Smart Eye : Using Pi Camera for Campus

Abstract. Home security has become the prime concern for everyone in recent times. Current systems available are unable to prevent the unwanted activities from happening. Expensive setup and high maintenance are common characteristics of these systems. In this paper a system is proposed to keep track of all the activities happening in the vicinity of the house. An attempt has been made to develop a home security system which is accessible, affordable and yet effective.
This paper describes a modification on the systems based on CCTV cameras to achieve the ability for keeping the target location (Home, Office, School etc.) safe from unwanted activities by implementing IoT technologies. The system proposed in this paper utilises the multi functionality of Raspberry pi such that after adding a camera module, it works as a security system. Raspberry pi comes with enough processing power to implement both face detection and recognition and hence used for the same motive. These component are implemented together in a way to minimise human intervention.
Proposed system has major functionalities like detecting the changes in the surveillance area and sending notification accordingly, face detection, live video streaming and an alarm system. Proposed system utilises functionalities of Raspberry pi and Python.

Keywords: Internet of Things (IoT), Raspberry Pi, AdaBoost, Local Binary pattern (LBP), Pi NoIR camera, Face detection, Histograms, FTP server

1 Introduction

Increased frequency of theft and burglary is a major concern for city dwellers. The facilities provided by the current system are not enough to prevent the crime from happening. The current system doesn't have a feedback loop to keep the concerned person aware of all the activities. Secondly, it is not capable in itself to detect any misconduct. Similarly high cost of installation make the concerned person with limited budget reluctant to have any kind of security system. As a result large portion of the city end up being a target area for thieves and mischiefs. Hence it is very important to upgrade the security level whether it is at home, office or school and to make it affordable.

Recent advancements in communication technologies or more specifically in IoT (Internet of Things) have opened a large spectrum of opportunities in the field of real-time systems and streamline processes.

These advancement have direct application in the security sector. With these advancements the user now can get the notification instantly which ultimately empowers the user to take measures proactively. With latest technology coming in handy for real time applications, an effective, accessible and affordable home security system is need of the hour.

The internet of things, or IoT, is a system of interrelated computing devices with the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. IoT devices share the data they collect by connecting to an IoT gateway or other edge device where data is either sent to the cloud to be analysed or analysed locally. Sometimes, these devices communicate with other related devices and act on the information they get from one another. The devices do most of the work without human intervention, although people can interact with the devices – for instance, to set them up, give them instructions or access the data.

2 Related Works

The Internet of things (IoT) is a system of interrelated computing devices, mechanical and digital machines provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. Such devices can be modelled into a security system. Steven Fernandez et al. [1], suggested a face detection system using Raspberry Pi. However authors did not approach face recognition implementation due to complexity of the recognition process, since recognition process would need more powerful resources to accomplish better results. Fernan Gallego et al. in [2] were able to switch from the closed-circuit television CCTV graphical processor to a computer graphical processing unit GPU, and embed the security cameras into the computer GPU. “Smart Eye” uses Raspberry pi zero W board which can perform both face recognition and detection. It also uses Raspberry Pi NoIR Camera Module v2 with a high quality 8 megapixel Sony IMX219 image sensor custom designed add-on board for Raspberry Pi.

Ping Hsin lee et al. in [3], attempted to detect faces in a digital image using various techniques such as skin colour segmentation, morphological processing, template matching, Fisher linear discriminant (FLD), and Eigen face decomposition. P. Viola et al. in [4] discussed rapid object detection method using a boosted cascade of simple features. There are three main contributions of object detection framework. The first contribution of this paper is a new image representation called an integral imagethat allows for very fast feature evaluation.The second contribution of this paper is a method for constructing a classifier by selecting a small number of important features using AdaBoost. The third major contribution of this paper is a method for combining successively more complex classifiers in a cascade structure which dramatically increases the speed of the detector by focusing attention on promising regions of the image.

Face recognition is one of the most important applications of image analysis, pattern recognition and image processing. Especially in the past two decades, many researchers have focused on it to find ways to improve the accuracy of recognition. Identification and verification are two main use of face recognition. General identity verification, Image database investigations, "Smart Card" applications and video indexing are some of applications of these two purpose.

V. Liong et al in [5] suggest LBP as a solution. Local Binary Pattern (LBP) is known as a powerful tool for texture classification and face recognition due to lack of sensitivity to light variation. Local binary pattern histogram (LBPH) calculates the histogram of all the LBP images. Then compares the histogram vector of each test images with all the train images and finds the best fit. Gray Level Co-occurrence Matrix (GLCM) has already been used for a face recognition system and it has good accuracy on ORL database. The result of direct GLCM have been compared with the results of Haralick's features and it was found the direct GLCM is more accurate for face recognition system. The co-occurrence matrix of local average binary pattern has also been used for face recognition system. In this technique, first, the average of gray-values of pixels are computed and then LBP operator is applied. Then, co-occurrence matrix of processed image is computed. Databases that are used to evaluate the algorithms have different features such as illumination variation, rotation, age variation and so on.

3 Hardware Implementation

Figure 1 is the Architecture Diagram of the System, the major components shown in this figure are Raspberry Pi NoIR camera Module, Raspberry Pi Zero W Board and User.

(I) LIST OF HARDWARE USED

- Raspberry pi zero W
- Raspberry pi NoIR camera Module
- Ada fruit raspberry pi zero w camera cable

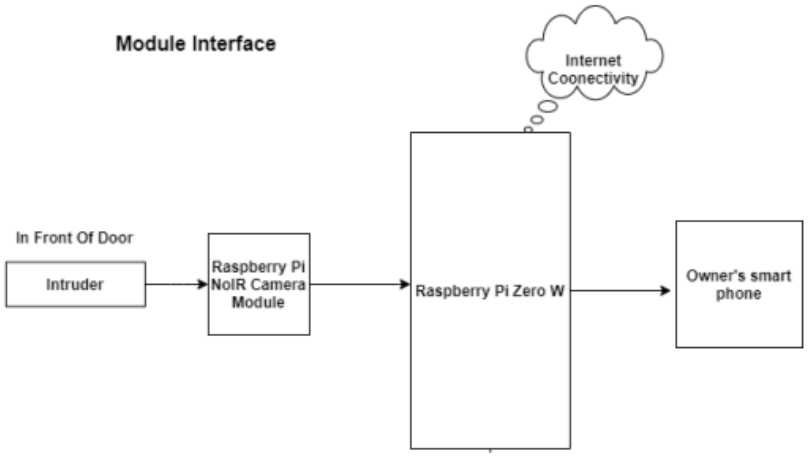

Fig. 1. Architecture diagram of the System

The architecture diagram describes the system as whenever any intruder arrives in front of door then system capture the frames of detected face by Raspberry pi NoIR Camera module which is attached with raspberry pi Zero W and process the image by comparing it with family member's image.
When system detects any difference between the images, it will send that image to owner's smart phone which should be connect through the internet.

4 Software Implementation

4.1 Algorithm

Face Detection

- Face detection is a technique that identifies or locates human faces in digital images.
- Face detection is different from Face recognition.
- Face detection detects merely the presence of faces in an image while facial recognition involves identifying whose face it is.
- Face detection is performed by using classifiers.
- A classifier is essentially an algorithm that decides whether a given image is positive(face) or negative(not a face).
- OpenCV already has two pre-trained face detection classifiers, which can readily be used in a program.
- The two classifiers are:
  i Haar Classifier

ii Local Binary Pattern classifier.

Local Binary Pattern classifier Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by threshold the neighbourhood of each pixel and considers the result as a binary number.

1. **Parameters:** The LBPH uses 4 parameters
   - **Radius**:The radius is used to build the circular local binary pattern and represents the radius around the central pixel. It is usually set to 1.
   - **Neighbours**:The number of sample points to build the circular local binary pattern. Keep in mind: the more sample points you include, the higher the computational cost. It is usually set to 8.
   - **Grid X**: The number of cells in the horizontal direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.
   - **Grid Y**: The number of cells in the vertical direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8
2. **Training the Algorithm**
   First, training of the algorithm is done. To do so, a dataset is used with the facial images of the people we want to recognize. Then set an ID (it may be a number or the name of the person) for each image, so the algorithm will use this information to recognize an input image and give an output. Images of the same person must have same ID.
3. **Applying the LBP operation**
   The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. The algorithm uses a concept of a sliding window, based on the parameters radius and neighbours.
   Figure 2 shows intermediate image of the original image in which decimal value of the generated binary number is then used for labeling the given pixel.

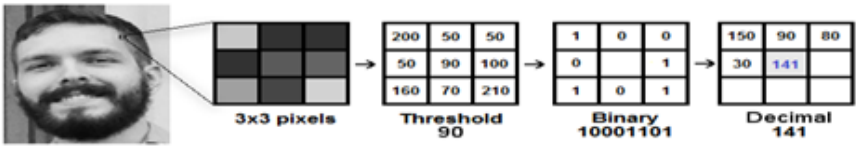

Fig. 2. Intermediate image of the original image

Figure 3 shows pixels from captured image.Based on the image above, breaking functions into several small steps make it comprehensive:

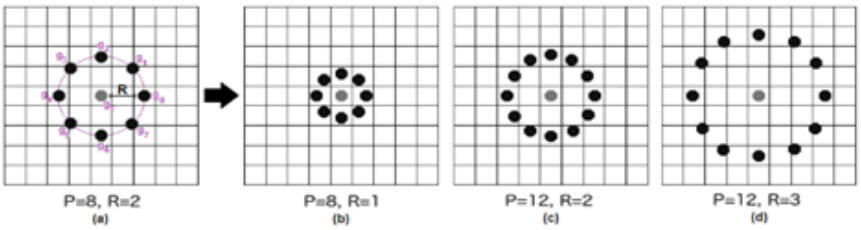

Fig. 3. Represent pixel from the original image

- Suppose there is a facial image in gray-scale.
- Get part of this image as a window of 3x3 pixels.
- It can also be represented as a 3x3 matrix containing the intensity of each pixel (0 255).
- Then, take the central value of the matrix to be used as the threshold.
- This value will be used to define the new values from the 8 neighbours.
- For each neighbour of the central value (threshold), set a new binary value. Set 1 for values equal or higher than the threshold and 0 for values lower than the threshold.
- Now, the matrix will contain only binary values (ignoring the central value). Concatenate each binary value from each position from the matrix line by line into a new binary value (e.g. 10001101). Note: some authors use other approaches to concatenate the binary values (e.g. clockwise direction), but the final result will be the same.
- Then, convert this binary value to a decimal value and set it to the central value of the matrix, which is actually a pixel from the original image.
- At the end of this procedure (LBP procedure), there is a new image which represents better the characteristics of the original image.

4. Extracting the Histograms

Now, using the image generated in the last step, use the Grid X and Grid Y parameters to divide the image into multiple grids.
Figure 4 shows histogram representation of the images as can be seen in the above image Based on the image above, extract the histogram of each region as follows:

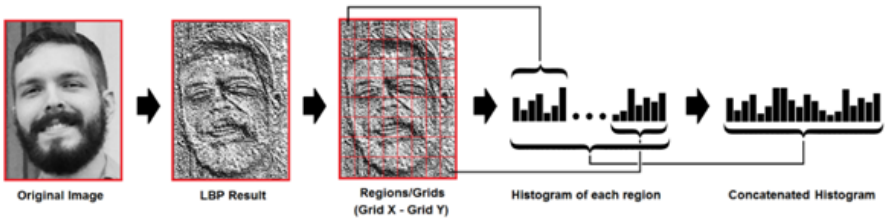

Fig. 4. Histogram of the original image

- There is an image in gray-scale, each histogram (from each grid) will contain only 256 positions (0 255) representing the occurrences of each pixel intensity.
- Then, concatenate each histogram to create a new and bigger histogram. Supposing we have 8x8 grids, we will have 8x8x256=16.384 positions in the final histogram. The final histogram represents the characteristics of the image original image.

5. Performing the Face Recognition

In this step, the algorithm is already trained. Each histogram created is used to represent each image from the training dataset. So, given an input image, perform the steps again for this new image and create a histogram which represents the image.

- So to find the image that matches the input image we just need to compare two histograms and return the image with the closest histogram.
- Various approaches can be used to compare the histograms (calculate the distance between two histograms), for example: **euclidean distance, chi-square, absolute value**, etc. In this example, the Euclidean distance is used (which is quite known) based on the following formula:

$$D = \sqrt{\sum_{i=1}^{n}(hist1_i - hist2_i)^2}$$

- So the algorithm output is the ID from the image with the closest histogram. The algorithm should also return the calculated distance, which can be used as a **'confidence'** measurement. **Note:** don't be fooled about the 'confidence' name, as lower confidences are better because it means the distance between the two histograms is closer.
- Then use a threshold and the 'confidence' to automatically estimate if the algorithm has correctly recognized the image. Assume that the algorithm

has successfully recognized if the confidence is lower than the threshold defined.

5 Result

To check system for different tasks, different types of test cases are prepared in order to verify system compliance, for example: 90 degree view of face and differentiating between family members and intruders etc. It was found that system is defect free and producing results as expected and well under permissible time delay.

6 Conclusion

The smart security facility developed under this project for home and other sensitive areas is manifested by using the available technologies. Hence they are easily available and are comparatively cheaper than other security systems. At an affordable price, this will ultimately empower a common city dweller against any mischief or attempt of theft on their property. The system has achieved functionalities such as, detecting intruders and family members, sending notification and updating files in FTP server. The system detects changes within surveillance area and checks for intruders, if it is not a family member then it sends notification to the user to apprise about the situation. The whole process takes about 2-3 seconds, which are enough for taking any effective actions to stop an ongoing theft. The home owner will not remain a passive observer but will be able to act immediately to avert any misconduct on her property. These measures also works as a deterrent for mischief.

7 The References Section

Steven Fernandez et al. [1], suggested a face detection system using Raspberry Pi which is compatible for Smart Eye. Fernan Gallego et al. in [2] embedded the security cameras into the computer GPU, but were unable to recognise faces.. "Smart Eye" uses Raspberry pi zero W board which can perform both face recognition and detection.Ping Hsin lee et al. in [3], attempted to use Fisher linear discriminant (FLD), and Eigen face decomposition whereas P. Viola et al. in [4] discussed rapid object detection method using a boosted cascade of simple features to detect face. Smart Eye uses LBP classifier for reasons such as time and resource constraints. V. Liong et al in [5] suggested LBP as a solution for face recognition. Given the constraints, this algorithm is best suited for Smart Eye.

References

1. V Swathi and Steven Fernandes. "Raspberry Pi Based Human Face Detection". In:Journal ofAdvanced Research in Computer4.9 (2015).

2. V. B. Saiz and F. Gallego. “GPU: Application for CCTV systems”. In:2014 International CarnahanConference on Security Technology (ICCST). 2014, pp. 1–4.
3. Ping Hsin Lee, Vivek Srinivasan, and Arvind Sundararajan.Face Detection, Final Year Project. 2014.
4. Paul Viola and Michael Jones. “Rapid object detection using a boosted cascade of simple features”.In:Proceedings of the 2001 IEEE computer society conference on computer vision and patternrecognition. CVPR 2001. Vol. 1. IEEE. 2001, pp. I–I.
5. J. Lu et al. “Learning Compact Binary Face Descriptor for Face Recognition”. In:IEEE Transactionson Pattern Analysis and Machine Intelligence37.10 (2015), pp. 2041–2056