iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** XII **Month of publication:** December 2023

DOI: https://doi.org/10.22214/ijraset.2023.57380

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue XII Dec 2023- Available at www.ijraset.com

Smart CCTV using Python and Open CV

Prof. Vanita Buradkar[1], Rizwan Khan[2], Arpita Satpute[3], Suraiya Pathan[4], Shraavi Ramteke[5]
Dept of Computer Science and Engineering, Rajiv Gandhi College of Engineering Research and Technology, Chandrapur, Maharashtra, India

Abstract: ***The field of video surveillance has seen significant advancements in technology. One such advancement is the incorporation of artificial intelligence (AI) into CCTV cameras, also known as "smart" CCTV cameras. These cameras have the ability to perform complex analytics that can greatly enhance the effectiveness of video surveillance. One of the key benefits of smart cameras is their ability to accurately detect and recognize objects, people, and events. This is achieved through the use of advanced algorithms that enable the camera to analyze video footage in real-time and identify any relevant activity.***
This is a python GUI application which can run on any operating system, uses webcam and has number of features which are not in normal CCTV. Smart CCTV camera here is used in conjunction with other technologies, such as facial recognition software, to identify individuals and track their movements [1].

I. INTRODUCTION

To create a python gui application that transforms normal cctv cameras to smart cctv camera by help of artificial intelligence and Logistic Regression classifier. Intelligent cameras are the name given to new types of surveillance cameras. The behaviour filtering capabilities of these cameras allow the camera to detect software with suspicious behaviour. While this situation minimizes the long and lost hours in terms of completely normal activities, it ensures that immediate action is taken. The proposed work overcomes this limitation by proposing supervised learning technique using Logistic Regression classifier. The results obtained are encouraging and show that the proposed system performed better than the similar methods. Nowadays, security is measure concern in every organization. To this satisfy issue the organizations use surveillance cameras. The limitation in using them is that there must be an operator to watch the stream from the cameras and take respective decisions. The use of camera based surveillance has extended from security to tracking, environment and threat analysis and many more. By using the power of modern computing and hardware it is possible to automate the process. The emergence of machine learning, Deep learning, and computer vision tools have made this process efficient and feasible for general purpose use[2].

II. EXISTING SYSTEM

Security cameras are widely used today. However, there is often a person who watches and interprets security cameras at any time. There is a need for systems that process the images on the security cameras and take the necessary precautions. Most of the times when an incident has already occurred we have to manually retrieve the saved footages and then manually check for suspicious activities[1].

A. Disadvantages Of The Existing System

1) There is no automated live detection its only post manual detection by a human
2) No face recognition ,no stolen object alert, no noise detection either
3) There's a lot of effort that's taken for monitoring hours long footages, this is not very convenient

III. PROPOSED SYSTEM

In this paper, we have created an software application that would transforms normal cctv cameras to smart cctv cameras by help of artificial intelligence and Logistic Regression classifier using python programming language and open cv as main module. The system proposed has monitoring feature for detecting stolen objects by smartly analysing frames and find the stolen objects by Structural Similarity. Face recognition feature ,This feature allows to train the model on known circle of us. Uses LBPH Face Identification method. Motion detection to a custom selected restricted area by It basically do absolute difference between two consecutive frames. From that difference we detect for boundaries. If boundaries are 0 then no motion and vice-versa. Feature to take picture and save it when someone enters and leave from the camera's frame[6]

©IJRASET: All Rights are Reserved

A. Advantages Of The Proposed System

1) Easy and cost free to implement
2) Plenty of people who use python and open cv package to help you out online
3) Reduced burden of constant monitoring by humans
4) Accuracy is better[5]

IV. MODULES

A. Trainingour Data

For testing purposes we decided to create our own training set by using a python script to download images from Google. We downloaded pictures of human walking. These images were kept in a separate folder for ease of access. After we had about 200 images it was time to make a weights file and a configuration file to use it. We made another script to generate xml files for each image. These xml files contained the exact area (or bounding area) in rectangle where the subject appeared to be performing its activity. The application was simple and every bounding box was required to be drawing over each image to generate the xml files After the xml files were generated we used a tiny OPEN CV]-test-voc file available on the official Open cv website. This small file is suggested to be used for training of custom object detection, using their CLI commands we trained the cfg file to detect our custom objects (i.e. activities of humans, walking in this case) the training of dataset was thus complete and it was then time to program the working of this.

B. Implementing Action Recognition

In the program the same network had to be fed 2 different configuration files. A recursive method was used for this. In our code we have 2 separate options for initiating the model. By using a second option we would use the same network with different configuration to process the activity in the function. We then code the program to search for the bounding box in the image. In a single image the various objects are detected like person. As a person is detected we can then call a subroutine or another function which will use the bounded area to detect what activity is done here. The difference between the two models is that the OPEN CV searches for what object is in the picture while this second function will try to understand what exactly lies in the marked area. As we see in the above image a human is detected. We send this sub image from the image into the next function which uses the same OPEN CV but with different weights and compares the activity dataset to recognize what activity is being done[10].

C. Monitor Feature

This feature is used to find what is the thing which is stolen from the frame which is visible to webcam. Meaning It constantly monitors the frames and checks which object or thing from the frame has been taken away by the thief. This uses Structural Similarity to find the differences in the two frames. The two frames are captured first when noise was not happened and second when noise stopped happening in the frame.

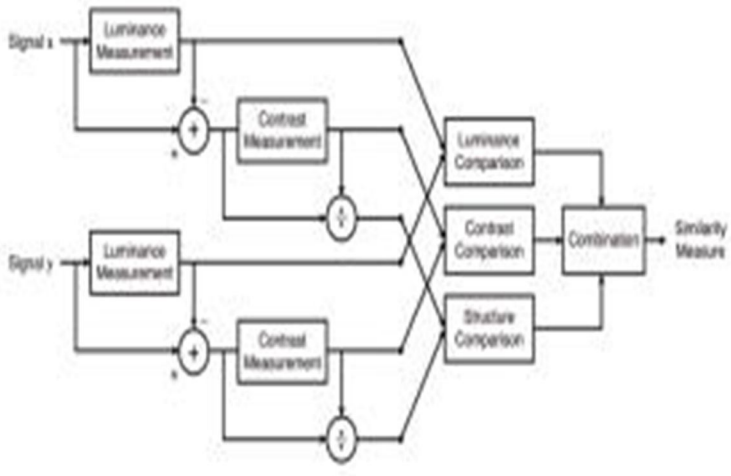


Fig 4.1.1 The comparison between the two images is performed

SSIM is used as a metric to measure the similarity between two given images. As this technique has been around since 2004, a lot of material exists explaining the theory behind SSIM but very few resources go deep into the details, that too specifically for a gradient-based implementation as SSIM is often used as a loss function. The Structural Similarity Index (SSIM) metric extracts 3 key features from an image:

1) Luminance
2) Contrast
3) Structure

The comparison between the two images is performed on the basis of these 3 features[10].

D. Identify The Member Feature

This feature is very useful feature of our minor project, It is used to find if the person the frame is known or not. It do this in two steps :

- Find the faces in the frames
- Use LBPH face recognizer algorithm to predict the person from already trained model. So lets divide this in following categories,

1) Detecting Faces in the Frames

This is done via Haarcascade classifiers which are again in-built in openCV module of python.

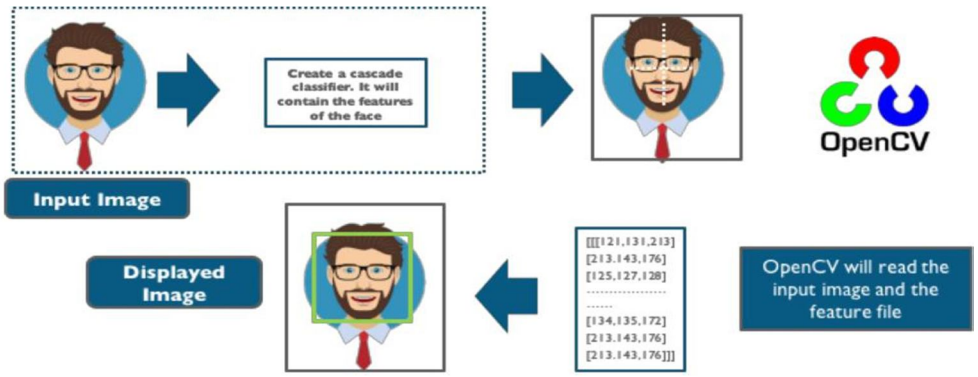


Fig 4.1.2 The working of open cv module

Cascade classifier, or namely cascade of boosted classifiers working with haar-like features, is a special case of ensemble learning, called boosting. It typically relies on Adaboost classifiers (and other models such as Real Adaboost, Gentle Adaboost or Logitboost).

Cascade classifiers are trained on a few hundred sample images of image that contain the object we want to detect, and other images that do not contain those images.

2) Using LBPH for Face Recognition

So now we have detected for faces in the frame and this is the time to identify it and check if it is in the dataset which we've used to train our lbph model.

The LBPH uses 4 parameters:

a) *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.
b) *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.
c) *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.
d) *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[8].

E. Inout Feature

This is the feature which can detect if someone has entered in the room or gone out. So it works using following steps:

1) It first detect for noises in the frame.
2) Then if any moiton happen it find from which side does that happen either left or right.
3) Last if checks if motion from left ended to right then its will detect it as entered and capture the frame. Or vise-versa.

So there is not complex mathematics going on around in this specific feature. So basically to know from which side does the motion happened we first detect for motion and later on we draw rectangle over noise and last step is we check the co-ordinates if those points lie on left side then it is classified as left motion[10].

V. SYSTEM REQUIREMENTS

A. Hardware Requirements

1) PC/laptop with at least 4GB RAM, 250GB HDD, i3 7th gen/AMD
2) Hard Disk:128 GB
3) RAM :1GB
4) Processor: Pentium
5) Webcam with drivers installed

B. Software Requirements

1) OS requirements: Linux\windows 7-11\Mac OS 10.6 -12
2) Python IDE: PyCharm\visual studio code
3) Python Packages : openCv, skimage,, numpy tkinter [10]

VI. SYSTEM MODELS

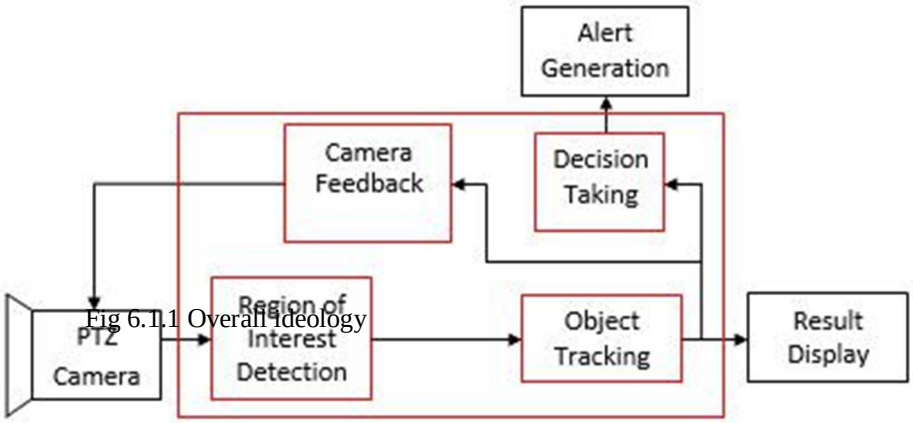
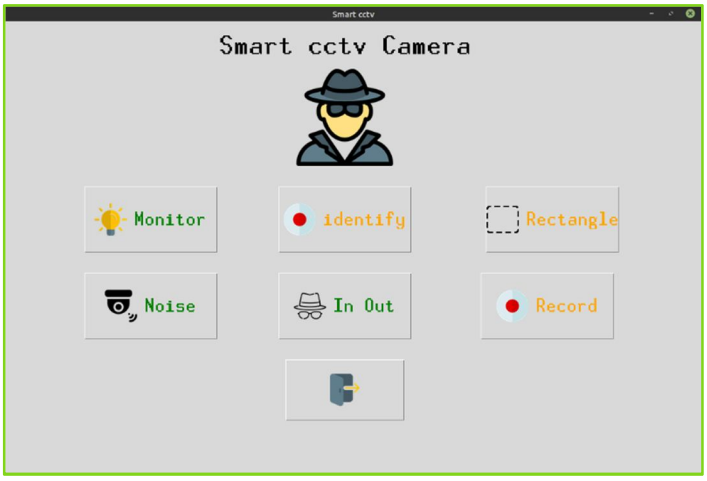


Fig 6.1.1 Overall ideology



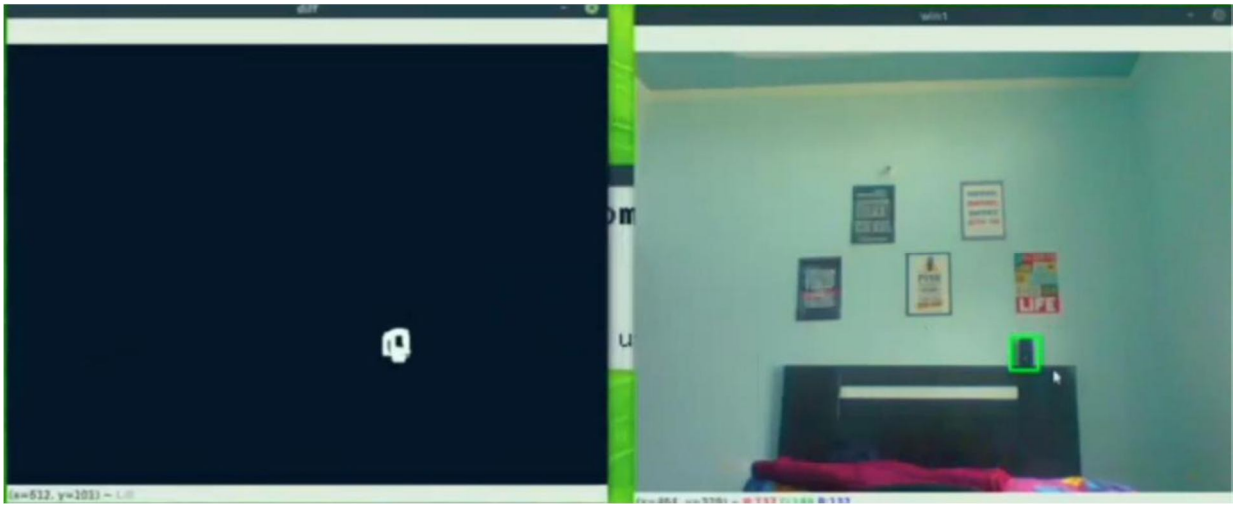


Fig 6.1.3 Monitor Feature - consider it as output from feature 1.
As you can see that it is detecting speaker is stolen which is true

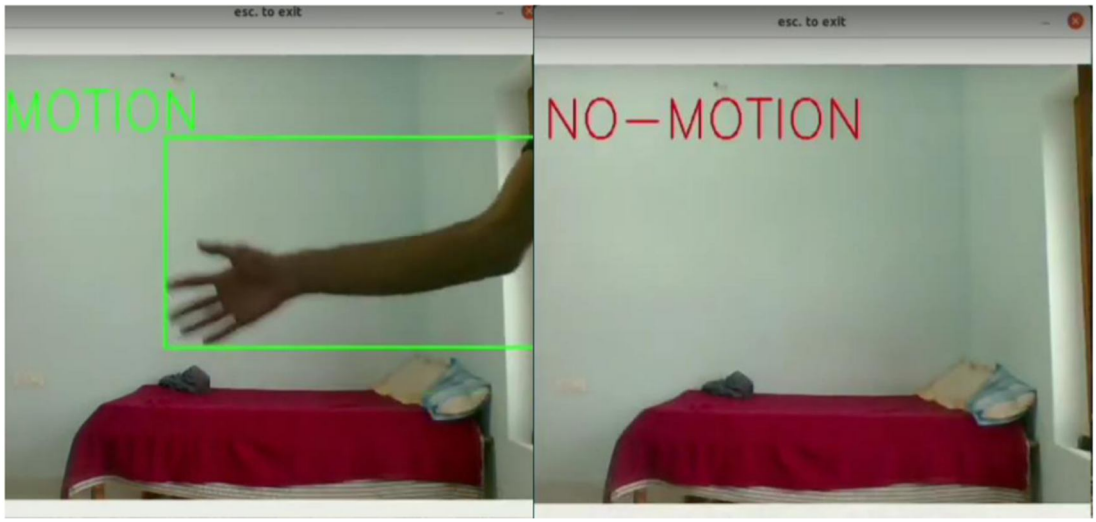


Fig 6.1.4 Noise Detection-This is working captured output for
NO-Motion and Motion being detected by this application.

VII. CONCLUSION

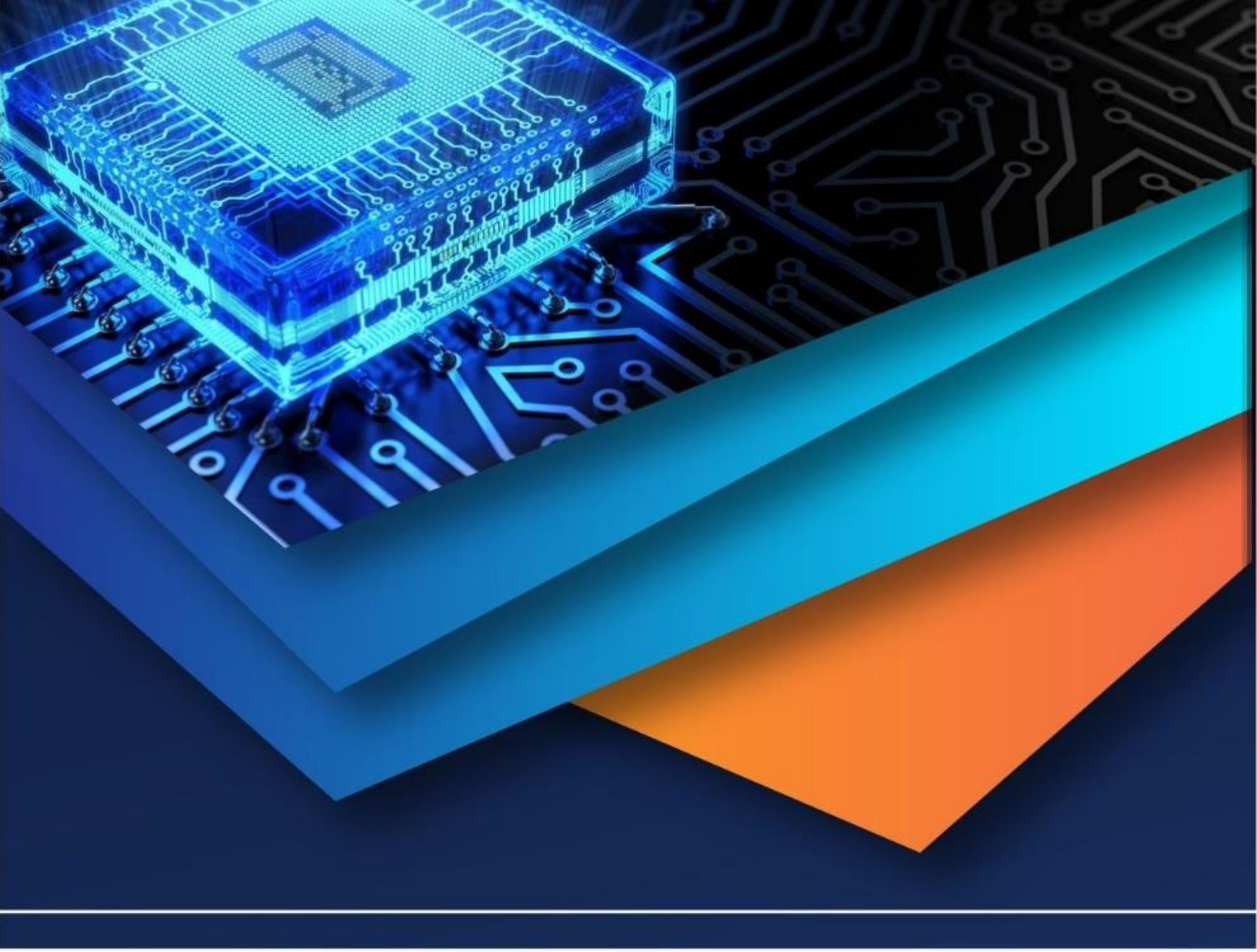
We found that for feature extraction and tuning system to work hand in hand with deep learning model haar cascade was useful. By using Image Mosaicing technique images could be stitched and camera position limitation were removed. Thus, among many methods of collecting camera input we found IP based camera technique on distributed network us useful and CNN model was useful for detail analysis.

This paper demonstrates and evaluates the usage of long term temporal convolutions and open cv for pattern matching, object detection and action recognition. Using space-time convolutions and OPEN CV over a large number of video frames, we obtain bounding boxed that detect the object or human in frame. With consequently larger training dataset, the media output will be much more efficient. Adding DL support would create broad scope in this project such as with DL we would be able to add up much more functionality[9].

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

REFERENCES

[1] Michael F.Adaramola ,Michael.A.K.Adelabu “Implementation of Closed-circuit Television (CCTV) Using Wireless Internet Protocol (IP) Camera” 1.School of Engineering, Lagos State Polytechnic, Ikorodu,P.M.B. 21,606, Ikeja. Lagos. Nigeria
[2] Tan Zhang, Aakanksha Chowdhery, Paramvir Bahl, Kyle Jamieson, Suman Banerjee “The Design and Implementation of a Wireless Video Surveillance System” University of Wisconsin-Madison, Microsoft Research Redmond, University College London
[3] C M Srilakshmi1, Dr M C Padma2 “IOT BASED SMART SURVEILLANCE SYSTEM” International Research Journal of Engineering and Technology (IRJET) Volume: 04 Issue: 05 | May -2017
[4] Mrs. Prajakta Jadhav1, Mrs. Shweta Suryawanshi2, Mr. Devendra Jadhav3 “Automated Video Surveillance eISSN: 2395 -0056 | p-ISSN: 2395-0072” International Research Journal of Engineering and Technology (IRJET) Volume: 04 Issue: 05 | May -201
[5] Pawan Kumar Mishra “A study on video surveillance system for object detection and tracking” Nalina.P, Muthukannan . K
[6] Jian Liang “Camera-Based Document Image Mosaicing” 18th International Conference on Pattern Recognition (ICPR'06)
[7] Akanksha Rastogi, Abhishesh Pal, Beom Sahng Ryuh “Real-Time Teat Detection using Haar Cascade Classifier in Smart Automatic Milking System” 2017 7th IEEE International Conference on Control System, Computing and Engineering, 24–26 November 2017, Penang, Malaysia
[8] Paul Viola, Michael Jones “Rapid Object Detection using a Boosted Cascade of Simple Features” 2001 IEEE
[9] Rahul Chauhan, Kamal Kumar Ghanshala, R.C Joshi “Convolutional Neural Network (CNN) for Image Detection and Recognition” 2018 IEEE
[10] https://youtu.be/L1ZQzdT4mEA?si=evLSTGnnCRnN4UJu



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)