International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 05 | May - 2022 Impact Factor: 7.185 ISSN: 2582-3930

# WatchDog- A Smart Surveillance System

**Kartikey Vaishnav[1], Asst. Prof. Mr. Ramesh Vaish[2] , Rajat Srivastava[3] , Nilesh Kumar Gupta[4] , Nikhil Kumar Choudhary[5]**

*1,3,4,5 Students, Department of Computer Science and Engineering, Babu Banarasi Das Institute of Technology & Management, Lucknow, India (formerly BBDNITM)*

*2 Assistant Professor, Department of Computer Science and Engineering, Babu Banarasi Das Institute of Technology & Management, Lucknow, India (formerly BBDNITM)*

---

## Abstract

In the modern world, security and privacy are the major concerns. Due to this, security cameras are installed in all public places to monitor any suspicious activities. But the general public does not know who is behind the screen watching them, monitoring their every move. It raises both securities as well as privacy issues. Another major issue regarding security is CCTV cameras are installed in public places only. Everyone cannot afford to hire someone to monitor their homes and alert them. This makes people at risk even in their homes. These issues lead to the development of WatchDog. WatchDog is a computer vision-based smart camera surveillance system that can monitor any suspicious activity and warn the owner about it. It uses python as its base language and uses OpenCV and skimage.metric library to capture any movement within its scope of vision. It is a simple yet effective program that can resolve many issues regarding security and privacy.

***Key Words***: Python, OpenCV, Smart-Surveillance System

## INTRODUCTION

In today's world, we have rising criminal activities like theft, vandalism, etc. it becomes necessary to install a 24*7 surveillance cameras at every place. So, WatchDog is a smart intrusion detector that analyzes the incoming video feed and raises the alarm in case of any suspicious activity. To achieve such functionality, we implemented four major modules

- Monitor
- Identification
- Motion Detector
- Entry-Exit.

The monitor module monitors the area and raises an alarm when anything goes missing. This module can be used to deter Anti Theft.

Identification modules tell about the person's name who has entered the area where WatchDog is installed. This module can be used for Authorization in a specific area.

The Motion Detector module detects any motion in the given frame and alerts the user. This module can be used to avoid Vandalism.

The EntryExit module keeps track of the number of people entering and exiting the facility. It can be used on both small and large platforms.

## 2. LITERATURE REVIEW

Field of Computer Vision have seen lot of boost in last decade due to availability of low cost hardware such as camera., which led to several previous surveys:

- Nishu Singla [1] proposed motion detection by finding frame difference methods. In this approach, detection of moving objects is found by finding differences between existing frame and reference frame.
- In the paper published by Ashish Kumar Sahu , Abha Choubey [2] they studied algorithms for Human Motion detection and also researched about the advantages and disadvantages of those algorithms. In particular, they explained background modeling, background Subtraction, characteristic robust motion detection, feature extraction.
- Arun Hampapur, Lisa Brown [3] featured applications and range of smart monitoring. They talked about a basic and active distributed monitoring architecture.
- Ramadan TH. Hasan, Amira Bibo Sallow [4] described various OpenCV algorithms and methods like Haar Cascade, LBP, LBPH, EigenFaces for face detection and recognition.
- Alessandro Massaro, Valeria Vittia [5] investigates the effectiveness of motion detection in video surveillance systems and the various factors that affect it.
- Mauricio Marengoni, Denise Stringhini [6] presented various methods for enhancing computer vision using OpenCV. It also described various advanced features of computer vision like tracking , optical flow and parallel computer vision.
- Shafie et al [7] presented motion detection using optical flow method. Optical flow can be caused by the relative motion of objects and the viewer, and it can provide useful information about the spatial arrangement of the things being observed as well as the rate at which this arrangement changes. Optical

flow discontinuities help with grouping Photographs of the area corresponding to various items.

- Ming-Hsuan Yang, David J. Kriegman, Narendra Ahuja [8] described in their investigation of facial recognition in images and studied more than 150 research paper and methods they described and categorized them in four categories: Knowledge-Based Top-Down Method, Bottom-Up Feature-Based Methods, Template Matching, Appearance-Based Methods

## PROBLEM STATEMENT

With rising incidents of criminal activity like Property theft, Vandalism, etc., it becomes necessary to install surveillance cameras. It is next to impossible to watch video feed 24*7 in the fast growing world.

Aim is to build a Smart Intrusion detector aka WATCHDOG to analyse incoming video feed and raise alarm. To ensure it fulfills almost all needs, we need functionalites like Anti theft and Motion detection. Features like Face recognition allows easier authentication and also assists in implementing Entry-Exit log in real time.

With these proposed features we should be able to bridge the gap between a manned cctv and an unmanned cctv surveillance

## PROPOSED APPROACH

To address the Problem noted in our Problem Statement, we might want a platform which could offer us with a Powerful Image Processing Library, a Mathematical Library to work with Multi-Dimensional Arrays, and Support for Designing Graphical User Interfaces.
The languages which achieve all the above requirements are Python and C++. C++ being not platform-independent, Python becomes the obvious choice.
Python provides an OpenCV module which can deal with image capturing and filtering. Skimage is another powerful library of Python that can be used in real-time video and frame monitoring. It also has a Tkinter module to design graphical user interfaces.
In our Proposed approach, we design four major modules in our WatchDog: Monitor, Identification, Motion detector, Entry Exit. These Four modules should deal with all the concerns and issues mentioned in Problem Statement.

## METHODOLOGY

### 1. FRAME MONITORING:

Frame Monitoring contains two major files. These are:
- find_motion.py
- spot_diff.py

find_motion.py file have two major bool variables named as motion_detected and is_start_done. These two variables manage the flow of the program and initially both are set to False. At first, the find_motion function captures the reference frame which it will use if the object is stolen, then it gets into an infinite while loop.While in loop, it captures two more frames to detect motion. It finds the absolute difference between both frames and saves it in a diff variable. This Variable is converted to binary and then contour found using findContours.
Using the coordinate of contour, we find the area of all the contours and select the one whose area is greater than 25mm$^2$. This ensures that only valid contours are selected and noises are removed. If the number of contours which get selected are more than 5 then we know that motion is detected. This means something might be stolen. So we set motion_detected to True. Finally, when the number of contours gets less than 3, it means motion has stopped and we need to check if something is stolen. So, We wait for some time and then call spot_diff.py.
spot_diff() function receives two arguments frame1 and frame2. These frames are converted to black and white and then blurred a little. This blur is to reduce noise. Then, it is given to the structural_similarity function of the skimage module. This returns two things: score and frame difference.
- Score shows how similar both of the images are.
- Frame difference is the absolute difference of pixel values between both frames.

The frame difference image is converted to binary and contours are found. Valid contour of area greater than 50mm$^2$ is selected. If the selected contour is more than 0 then we mark it using a rectangle in the initial frame and return 1 else we return 0 and find_motion keeps on looping again. At last the missing object image is stored in storage and the date and time of stealing is saved in the log.
Figure a. shows the flowchart diagram of working of Frame Monitoring Module.

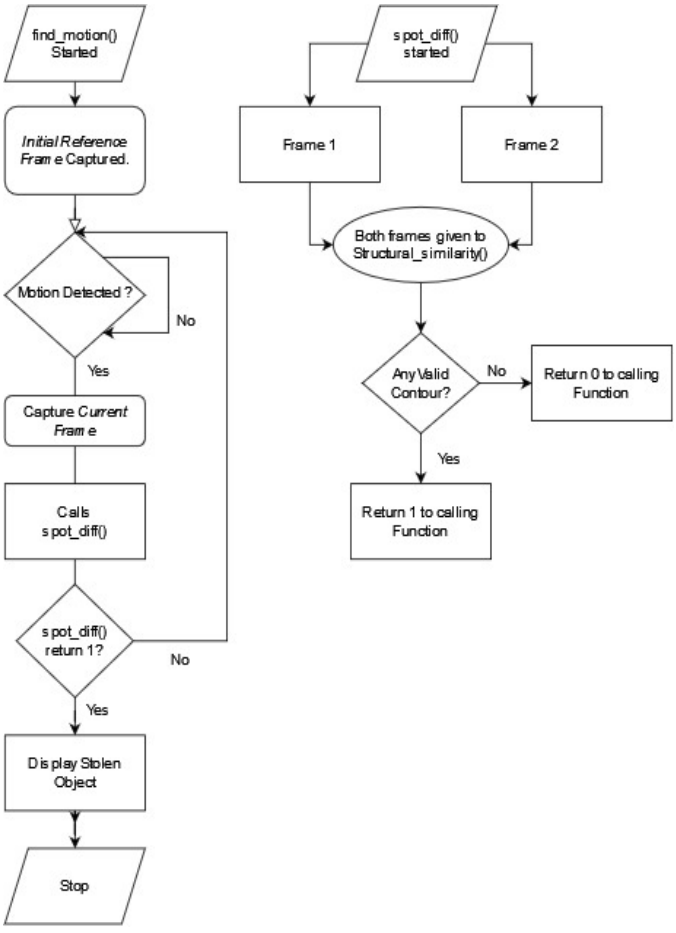

**Figure a.**

**2. IDENTIFICATION:**

It contains three major parts:
1. First is to find a person's face using HaarCascade which is inbuilt in OpenCV. We take 3 haar features which are line, edge, and rectangle, and run it across the image to find facial features. In function collect_data() ,we use OpenCV to capture a video feed of our objective (which is a person whose face is to be identified) and run HaarCascade to find coordinates across the face to make a rectangle to show that face is present in that area.
2. Second we call the train() function to train our LBPH(local binary pattern histogram) algorithm and we use a minimum of 300 images to train our 4 classifiers namely radius , neighbor , Grid x , Grid y . Now we create a histogram using Grid x and Grid y.
3. Now we call identify() function using our GUI.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 picture, we carry out the steps once more for this new image and create a histogram that represents the image. Now we evaluate the histogram by locating the absolute difference, which tells us whether or not the image matches or not.
Figure b. shows the flowchart diagram of working of Identification Module

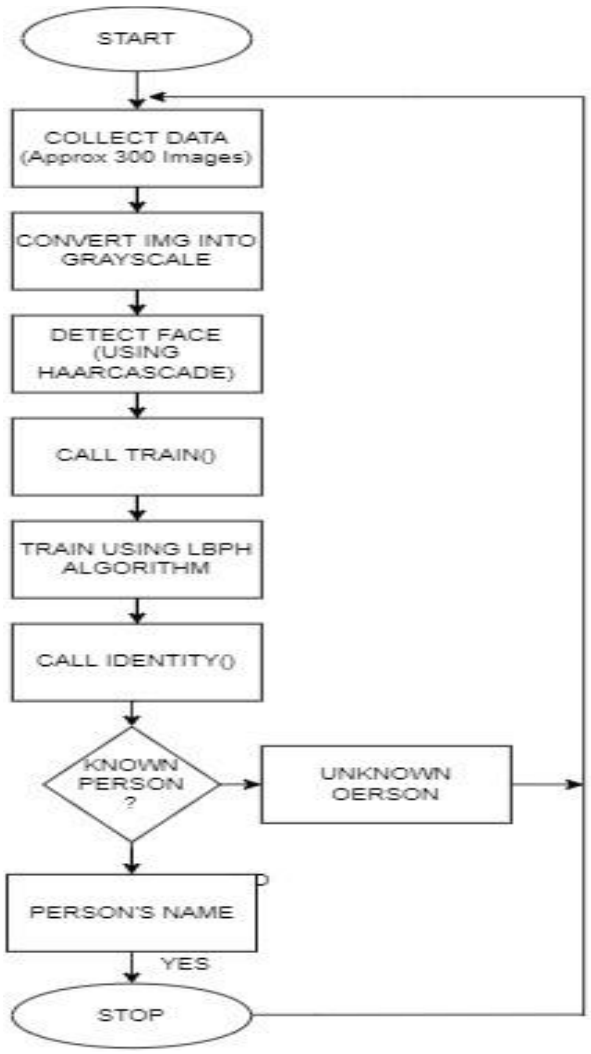


**Figure b.**

**3. MOTION DETECTION:**

Motion Detection contains one file:

- motion.py

At first, the noise() function captures two frames to detect motion.
It finds the absolute difference between both the frames and saves it in a diff variable.
This Variable is converted to binary and then contour found using findContours.
Using the coordinate of contour, we find the area of all the contours and select the one whose area is greater than 0mm^2. This ensures that valid contours are selected and noise detected and it will return "Motion Detected" and if contour is not detected then it will return "Motion Not Detected".
Figure b. shows the flowchart diagram of working of Motion Detection Module.bn

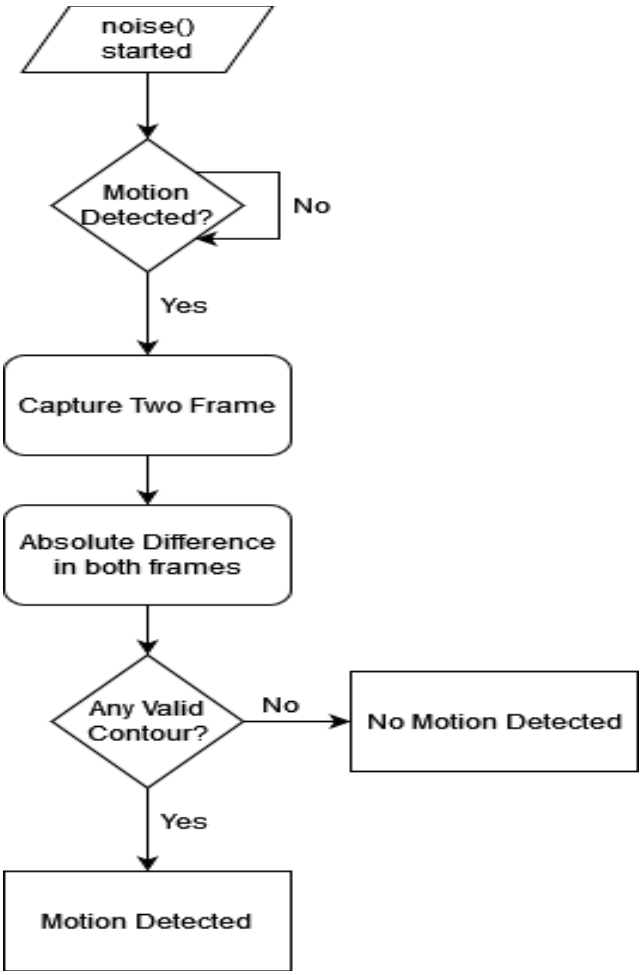


**Figure c.**

**4.ENTRY-EXIT:**

Entry-Exit contains following file:

- in_out.py

Frame 1 and Frame 2 that is the former and later frame are initialized, Flipped/ Mirrored cv2.flip to facilitate viewing is performed, Absolute difference between both the frames is calculated via "cv2.absdiff" to isolate moving elements which are stored in a variable.

Further, the image is blurred via "cv2.blur", Later on preprocessing of the image is finished by making it grayscale using cv2.cvtColor

Image is converted to absolute black and white using a given predefined threshold using cv2.threshold. This threshold facilitates the formation of contour.

To find a contour in the image we use "cv2.findContours" and provide "cv2.RETR_TREE" in the first argument to ensure successive attempts to find a complete contour and save the points using "cv2.CHAIN_APPROX_SIMPLE".

Once we have all the necessary variables for the process we use simple logic to detect whether the person came in or went out.

If our current contour exists with a viable length we find its max area and form an approximate binary rectangle around the image via cv2.boundrect

Again around our current rectangle which is internally loaded, we form a visible rectangle and we classify it as a recognizable motion.

Our Left and Right variables are initialized, now the net position of the object is provided on the basis of the approximate rectangle we built earlier. If its x edge is greater than a certain threshold (500 in our scenario) we classify the object as on the right side of the frame else it is left.

But if the object was already on the right and then it was next seen on the right side of the frame we mark it as the object going to the left and vice versa for the right side.

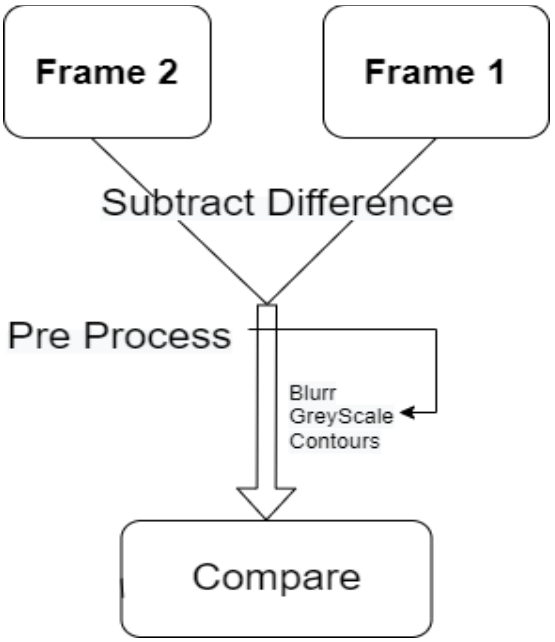


**Figure d.**

## FUTURE WORK

Watchdog is made to be open source and its code is publicly available for anyone who intends to use it or to tweak it. Being built on python entirely it is easy to understand and can be modified to serve more purposes. As of now, Watchdog is best suited for personal use but it has the high scope of being turned into much more than it. For reference, the watchdog can be modified to trigger the home cooling system on the arrival of members or similarly turn them on exit without requiring a manual command. Such home automation can prove to be game-changing as a significant power-saving mechanism. Moreover, it can be deployed in industrial facilities to completely automate surveillance and the manpower freed from it can be employed elsewhere to improve productivity. It can even be deployed on school premises to automate attendance. The possibilities for this technology are endless and we should harness it to the best of our abilities.

## CONCLUSION

Surveillance has become very important for people's security and it is impossible to have someone who can always observe the video feed, also there is always a margin for human error.

All the above issues make Computer Vision a viable option. Computer Vision is fast and reliable, also it costs minimal to none.

Even though it is impossible to completely remove human intervention in the near future, we can minimize it as much as possible.

WatchDog is one such possible option. Even though it is still in the development phase, it still gives promising results. In the near future, it can be made industry ready to be deployed both in homes and public places.

## REFERENCES

1. Nishu Singla, "Motion Detection Based on Frame Difference Method", International Journal of Information & Computation Technology. ISSN 0974-2239 Volume 4, Number 15 (2014), pp. 1559-1565.

2. Ashish Kumar Sahu , Abha Choubey, "Motion Detection Surveillance System Using Background Subtraction Algorithm", International Journal of Advance Research in Computer Science and Management Studies.ISSN: 2321-7782 Volume 1, Issue 6, November 2013.

3. Arun Hampapur, L. Brown, Jonathan Connell, S. Pankanti, Andrew Senior,Y. Tian "Smart surveillance: Applications, technologies and implications" DOI:10.1109/ICICS.2003.1292637, Conference: Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on Volume: 2

4. Ramadan TH. Hasan, Amira Bibo Sallow "Face Detection and Recognition Using OpenCV" Journal of Soft Computing and Data Mining Vol. 2 No. 1 (2021) p. 86-97

5. Alessandro Massaro, Valeria Vitti, Giuseppe Maurantonio,Angelo Galiano, "SENSITIVITY OF A VIDEO SURVEILLANCE SYSTEM BASED ON MOTION DETECTION ", Signal & Image Processing : An International Journal (SIPIJ) Vol.9, No.3, June 2018

6. Maur´ıcio Marengoni and Denise Stringhini, "High Level Computer Vision using OpenCV", 2011 24th SIBGRAPI Conference on Graphics, Patterns, and Images Tutorials

7. M. H. Ali, Fadhan Hafiz, A. A Shafie, Motion Detection Techniques using Optical Flow, World Academy of Science, Engineering and Technology, vol.32, pp 559-561, 2009.

8. Ming-Hsuan Yang, David J. Kriegman, Narendra Ahuja, "Detecting Faces in Images: A Survey", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 24, NO. 1, JANUARY 2002