

# Automatic Human Detection in Surveillance Camera to Avoid Theft Activities in ATM Centre using Artificial Intelligence

M. Baranitharan, R. Nagarajan, G. ChandraPraba  
AP/CSE

Computer Science and Engineering  
Kings College of Engineering, Thanjavur

**Abstract**— Nowadays most of the 300 million surveillance cameras today are ‘blind’ and merely record videos for post-incident manual analysis. So The system deals with the development of an application for automation of video surveillance in ATM machines and detect any type of potential criminal activities that might be arising with the automated system which would considerably decrease the inefficiency that are existing in the prevalent systems. An advanced Human detection system using Open Computer Vision technique and Artificial Intelligence would be utilized which would create phenomenal results in the detection of the activities and their categorization. The proposed system makes efficient utilization of Open CV which has more than 2500 optimized algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects finally ending up with the detection and identification of the necessary action for the prevention of such type of activities. The proposed system includes the specialized mechanisms for Camera tampering, Collision of human, Risky voice analysis, long time tracking. The entire mechanism takes place in real time decreasing the time complexity to a great extent making the system an efficient mechanism to prevent such anti-social activities.

## INTRODUCTION

It is a well-known fact that digital India is the outcome of many innovation and technological advancements. Nowadays Surveillance cameras in ATM centers are only for recording purpose. If any theft activities are occurred, it will be known only by human information. Then police will start investigating by the help of CCTV records. Some time “Thieves” will cover or destroy the camera so it can’t be able to record. The world scenario witnesses extensive usage of automated video surveillance systems which plays a vital role in our day to day lives in order to enhance protection and security for individuals and infrastructure. Tracking and detection of objects is an essential component in various traffic monitoring systems, biometrics and security infrastructures, safety monitoring, various web applications and recognition of objects for mobile devices etc. One major application area of this process is the detection of robbery. In this system the primary focus will be in the field of detection of suspicious activities or crime in an ATM (Automatic Teller Machine) which is basically a profitable bank service which enables financial transactions in

public spaces where the machines are a replication of the bank clerks and tellers. Although several researches are going on in the field of ATM crime detection, however the utilization of the crime detection system is scarcely observed due to lack of efficiency and processing in the existing crime detection systems. Hence the idea of creating such an automated system was conceived after relative observations of the real life incidents that are happening in and around the globe. The increasing proliferation of the ATM frauds which involves activities like Camera Covering, Money grabbing inside ATM Center, Stealing the ATM Machine, Risky Voice is a matter of concern which would be tackled by the proposed system to enable secure financial transaction at anytime.

## OPENCV

OpenCV (Open Source Computer Vision Library) is an open-source BSD-licensed library that includes several hundreds of computer vision algorithms. The document describes the so-called OpenCV 2.x API, which is essentially a C++ API, as opposite to the C-based OpenCV 1.x API

## PYTHON

Python is an interpreted high-level programming language for general-purpose programming. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library. Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations

## LIBRARIES IN PYTHON

Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited to many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary precision decimals, manipulating regular expressions, and unit testing.

Some parts of the standard library are covered by specifications (for example, the Web Server Gateway Interface (WSGI) implementation `wsgiref` follows PEP 333), but most modules are not. They are specified by their code, internal documentation, and test suites (if supplied). However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations.

- Graphical user interfaces
- Web frameworks
- Multimedia
- Databases
- Networking
- Test frameworks
- Automation
- Web scraping
- Documentation
- System administration
- Scientific computing
- Text processing
- Image processing

#### CAPTURE VIDEO FROM CAMERA

The System has to capture live stream with camera. OpenCV provides a very simple interface to this. Let's capture a video from the camera (using the in-built webcam of my laptop), convert it into grayscale video and display it.

To capture a video, you need to create a `VideoCapture` object. Its argument can be either the device index or the name of a video file. Device index is just the number to specify which camera. Normally one camera will be connected (as in my case). So I simply pass 0 (or -1). You can select the second camera by passing 1 and so on. After that, you can capture frame-by-frame. But at the end, don't forget to release the capture..

#### BACKGROUND SUBTRACTION

Background subtraction is a major preprocessing step in many vision-based applications. For example, consider the case of a visitor counter where a static camera takes the number of visitors entering or leaving the room, or a traffic camera extracting information about the vehicles etc. In all these cases, first you need to extract the person or vehicles alone. Technically, you need to extract the moving foreground from static background.

If you have an image of background alone, like an image of the room without visitors, image of the road without vehicles etc, it is an easy job. Just subtract the new image from the background. You get the foreground objects alone. But in most of the cases, you may not have such an image, so we need to extract the background from whatever images we have. It becomes more complicated when there are shadows of the vehicles. Since shadows also move, simple subtraction will mark that also as foreground. It complicates things.

#### OBJECT DETECTION

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones. Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones.

Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, Haar features shown in the below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle.

#### SPEECH RECOGNITION

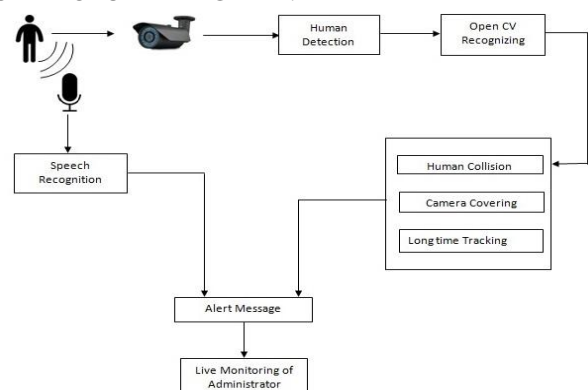
Google API Client Library for Python is required if and only if you want to use the Google Cloud Speech API (`recognizer_instance.recognize_google_cloud`).

If not installed, everything in the library will still work, except calling `recognizer_instance.recognize_google_cloud` will raise an `RequestError`.

According to the official installation instructions, the recommended way to install this is using Pip: execute `pip install google-api-python-client` (replace `pip` with `pip3` if using Python 3).

Alternatively, you can perform the installation completely offline from the source archives under the `./third-party/Source` code for Google API Client Library for Python and its dependencies/ directory.

#### ARCHITECTURE DIAGRAM:



#### MODULES

##### A. HUMAN DETECTION

When the human enters into the ATM center the Surveillance camera start tracking with the bounding box by the identification of human parts. Haar cascade is the library which works with Open CV to detect the human. This module will be helpful in Long time tracking and Human Collision.

##### Haar CASCADE

It will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then it needs to

extract features from it. The detection can be takes place by green rectangle bounding box.

### B. Long Time Tracking

The module takes input from the Human Detection. When the human enters into the system it calls the timer to measure the time. Start Stop timer is the unique library used for time measurement. When the predefined time limit for particular human detection is reached, the system sends the alert mail to the admin.

#### TIMER

Timers are started, as with threads, by calling their start() method. The timer can be stopped (before its action has begun) by calling the cancel() method.

### C. CAMERA TAMPERING

In this module, the status of the camera is considered. If lens is sprayed by glue and the image is darkened, it will not be possible to distinguish this from other situation where the same effect is seen, i.e. when the lighting conditions change. When this parameter is enabled, alert mail will be generated for all cases where the image turns dark or the lens is sprayed.

#### BACKGROUND SUBTRACTOR

It uses a method to model each background pixel by a mixture of K Gaussian distributions ( $K = 3$  to  $5$ ). The weights of the mixture represent the time proportions that those colours stay in the scene. The probable background colours are the ones which stay longer and more static.

While coding, it need to create a background object using the function, `cv2.createBackgroundSubtractorMOG()`. It has some optional parameters like length of history, number of gaussian mixtures, threshold etc. It is all set to some default values. Then inside the video loop, use `backgroundsubtractor.apply()` method to get the foreground mask

### D. HUMAN COLLISION

In this module the input is from the Human Detection. From the human detection module we know every human part is bounded by the respective bounding boxes and tracking is continued. If more persons are in the same environment, the multiple detection and tracking is also possible. So from that the system has multiple boxes in the environment. If the collision is occurs between the boundaries then the violations may happened, so the system sends the alert message to the admin.

#### LUCAS KANADE PYRAMID

OpenCV provides a single function, `cv2.calcOpticalFlowPyrLK()`. Here, it create a simple application which tracks some points in a video. To decide the points, we use `cv2.goodFeaturesToTrack()`. It take the first frame, detect some Shi-Tomasi corner points in it, then iteratively track those points using Lucas-Kanade optical flow. For the function `cv2.calcOpticalFlowPyrLK()` pass the

previous frame, previous points and next frame. It returns next points along with some status numbers which has a value of 1 if next point is found, else zero. It iteratively pass these next points as previous points in next step

### E. VOICE RECOGNITION

In this module the risky voice analysis can takes place. When the customer enters into the system the voice recognition is started. The API is converts the voice into text. The Risky voice strings are "Help" and "Emergency".

#### GOOGLE SPEECH API

It converts spoken text into written text (Python strings), briefly Speech to Text. Google API will translate this into written text. It has excellent results for English language.

### CONCLUSION

In this paper we implements Artificial Intelligence in surveillance camera with an advanced computer vision techniques it really helps to avoid the theft activities in ATM centers before they really acquired.

### REFERENCES

- Ing. Ibrahim Nahhas ,Ing. Filip Orsag, Ph.D, "Human Face Detection Using Skin Color Information," Bruno University of Technology. (*references*)
- Siti Nur Ateeqa Mohamad, Ahmad Ammar Jamaludin, Khalid, "Speech Semantic Recognition - A Control Mechanism For Assistive Robot," Universiti Tun Hussein Onn Malaysia (UTHM).
- Neeti A. Ogale" A Survey of Techniques for Human Detection from Video", University of Maryland, College Park, MD 20742.
- [4] Ing. Ibrahim Nahhas, Ing. Filip Orsag, Ph.D "Real Time Human Detection And Tracking", Bruno University of Technology.
- [5] Mohamed Hussein, Wael Abd-Elmageed, Yang Ran, Larry Davis "Real-Time Human Detection in Uncontrolled Camera Motion Environments" Institute for Advanced Computer Studies University of Maryland.
- [6] Wongun Choi, Caroline Pantofaru, Silvio Savarese "Detecting and Tracking People using an RGB-D Camera via Multiple Detector Fusion" Electrical and Computer Engineering, University of Michigan, Ann Arbor, USA.
- [7] Rupesh Mandal, Nupur Choudhury "Automatic video surveillance for theft detection in ATM machine : An enhanced approach", Computer Science and Engineering Sikkim Manipal Institute of Technology India.
- [8] Eun Som Jeon, Jong-suk Choi, Ji Hoon Lee, Kwang Yong Shin, Yeong Gon Kim, Toan Thanh Le And Kang Ryoung Park," Human Detection Based on the Generation of a Background Image by Using a Far-Infrared Light Camera", Division Of Electronics And Electrical Engineering, Dongguk University.