Smart CCTV: An Integrated approach of Computer Vision and Machine Learning for Advanced Surveillance

Abhay Chopde
Department of Electronics and Telecommunication Engineering
Vishwakarma Institute of technologyPune, India
abhay.chopde@vit.edu

Balganesh Thombre
Department of Electronics and Telecommunication Engineering
Vishwakarma Institute of technologyPune, India
balganesh.thombre21@vit.edu

Vedant Khodake
Department of Electronics and Telecommunication Engineering
Vishwakarma Institute of Technology,Pune, India
anil.vedant22@vit.edu

Nikhil Yeware
Department of Electronics and Telecommunication Engineering
Vishwakarma Institute of technologyPune, India
nikhil.yeware21@vit.edu

***Abstract*— The Smart CCTV is an improved version of the standard CCTV equipment, offering functions not seen in the modern surveillance systems, such as visitor verification, noise detection, and facial recognition. Furthermore, the Smart CCTV boasts a motion detection and alarm feature that notifies the user in the event of theft and signals the removal of any object from the frame.**

Keywords: Visitor verification, alarm feature, real time weapon detection, structural similarity index, LBPH algorithm, python GUI application, crime prevention, data analytics.

I. INTRODUCTION

The smart CCTV, is a type of security camera that monitors both public and private areas. In recent years, the demand for CCTV devices has risen sharply. They are now very helpful in homes, workplaces, and retail establishments. Installing a CCTV system is a sensible decision when it comes to family or business safety. Installing a CCTV camera alone, however, is insufficient; it also needs to contain functions that could be vital in an emergency. It must also be reasonably priced and require little upkeep. Having a CCTV system gives the victim an unfair edge if an unfortunate occurrence, like a crime, occurs at the location because it offers a means of gathering evidence that will aid in deduction and investigation. With additional evidence from a CCTV camera, helpful places, times, locations, and, most crucially, the ability to identify criminals, crimes can be solved much more quickly. Referring to CCTV footage, will save time and will be extremely convenient in many scenarios. Furthermore, CCTVs record and store 24-hour video footage of the area in which they are situated, which can contribute to a sense of security. Because of the assistance that CCTV cameras offer in a variety of these circumstances, its significance is thus well known. With the help of this project, a CCTV system that can watch and record video, prevent crimes from occurring, and quickly identify potential fleeing criminals will be made available.

II. LITERATURE REVIEW

Based on the study's findings, automatic weapon identification in real-time CCTV footage can be achieved with the use of deep learning algorithms, particularly YOLOv4. This can contribute to the maintenance of a secure and safe environment. The study comes to the conclusion that real-time use of deep learning-based techniques utilizing CCTV cameras may effectively detect weapons. The research comes to the conclusion that region proposal-based techniques can increase the accuracy and speed of detection and that deep learning can be applied well for real-time weapon identification in CCTV footage. Low-resolution CCTV streams can effectively be used by the suggested automatic weapon identification system to identify firearms utilizing deep learning in real time models. For real-time weapon identification, the YOLOv4 model is the best option[1].

As the suggested approach would eliminate the difficult manual procedure that law enforcement agencies currently use, it may make it easier to identify criminals and missing persons from CCTV footage. The results of the study suggest that the proposed system could be utilized by law enforcement agencies for searching for missing individuals as well as for criminal detection and tracking. It can be used for facial recognition and verification in photos and videos[2].

The proposed system is a reliable, cost-effective, real-time automated accident detection solution that requires minimal hardware installation. The suggested system detects automobiles, tracks their movement, and classifies them as damaged or undamaged with high accuracy and real-time performance. With the SVM-rbf classifier and Mini-YOLO obtaining high accuracy and low latency, the study successfully built an autonomous accident detection system that can be implemented in real-time scenarios[3].

The research comes to the conclusion that by automating the process of monitoring CCTV feeds, decreasing human error, and boosting the performance of surveillance systems, the Smart-Surveillance System has the potential to completely transform the surveillance industry. To create feasible applications, the system can be integrated with different video processing features[4].

The suggested architecture increases the detection rate in difficult PTZ and motion blur CCTV scenarios. Although the framework can be applied to any kind of object, pedestrians are the main focus of the studies. The proposed methods have been shown to enhance the performance of deep learning-based multi-target detectors when applied to challenging CCTV footage.[5].

III. METHODOLOGY

This Python GUI application works on any operating system. It employs a webcam and has additional features that are not seen in standard CCTV.

i. Monitor Feature

This function finds the object that was removed from the first webcam image. It constantly monitors the frame and checks for things that have moved within it. This function examines structural similarities to determine the difference between two frames. Structural similarity index (SSIM) measures extract three important aspects from images - *Contrast, Luminance and Structure.*

The two images are compared on the basis of these 3 features.

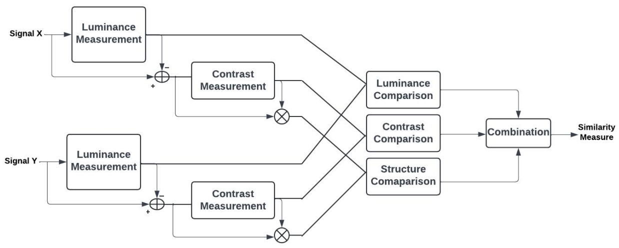

Fig.1. Structural similarity index (SSIM)

This system measures the similarity between two images by calculating the Structural Similarity Index (SSIM). The SSIM value ranges from -1 to 1. A score closer to 1 signifies high similarity, while a score closer to -1 indicates significant dissimilarity. For practical purposes, these values are often rescaled to a range of 0 to 1, where 0 represents complete dissimilarity and 1 represents perfect similarity.

Luminance is calculated by averaging over all pixel values. The formula, represented by μ (Mu), is as follows:

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i.$$

Structure: Structural comparisons are made using an integrated formula, but the input signal is divided by its standard deviation, resulting in a standard deviation of one unit, allowing for more robust comparison.

$$(\mathbf{x} - \mu_x)/\sigma_x$$

ii. Face Recognition

This functionality is used to determine whether a person is known within the program. There are two steps involved.

- Locating a face in a frame.
- Applying the LBPH detection algorithm to predict a person face from a trained model.

Face Detection using LBPH

Now that we've found faces in the picture, we'll identify them and determine if a person appear in the dataset used to test the LBPH algorithm model.

The LBPH uses following parameters.

• *Radius*: The distance from the center pixel needed to generate a circular local binary pattern.
• *Neighbors*: This is the number of sampling sites required to create the circular local binary pattern. It should be noted that increasing the number of sample points requires greater computer resources. Typically, eight sampling sites are employed.
• *Grid X:* Specifies the number of horizontal cells in the grid. More cells form a finer grid, resulting in a larger feature vector. The common setting is usually 8.
• *Grid Y* : Indicates the number of vertical cells in the grid. Similar to Grid X, more cells refine the grid and increase the size of the feature vector. This is likewise typically set to 8.

The initial phase of the LBPH algorithm involves creating a modified image that highlights facial features more prominently. This is accomplished by applying a sliding window technique guided by the radius and neighbor parameters. This process is visually depicted in the figure below.

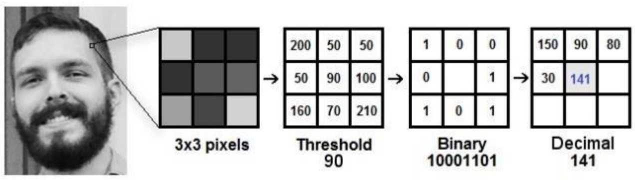

Fig.2. Creation of intermediate image highlighting facial traits

Extracting Histograms: The image obtained in the final stage is divided into numerous grids using the Grid X and Grid Y parameters, as illustrated in the image below:

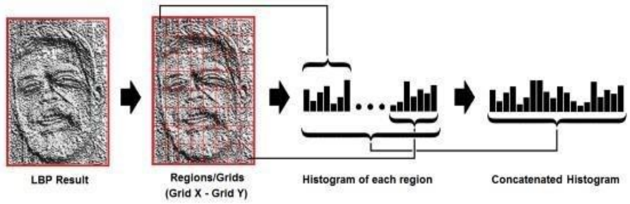

Fig. 3. Division of image into multiple grids

Following this, the model is trained to produce predictions, and the same methods are used to make and compare its histograms to the previously trained model, demonstrating how this feature works.

iii. Noise Detection

This feature is utilized to successfully detect image noise, which is seen on most CCTV. Basically, each frame is constantly evaluated and verified for noise. Noise is measured in consecutive frames. The absolute difference between the two frames is calculated, the picture difference is examined, and the contour (motion border) is identified. There is no movement if there are no boundaries, yet movement occurs when there are.

iv. Visitor Detection

This is a function that detects when someone enters or leaves the room. The following technique demonstrates how it works:

- First, it detects noise in the image.
- When motion is detected, it will determine which side the motion originated (left or right).
- Finally, detect movement and capture an image of the frame.

IV. TECHNOLOGY USED

The requirements to run this application are as follows:

• Any platform, including Windows, Linux, and Mac OS.
• Any python version available.
• Python packages include OpenCV, Skimage, NumPy, and Tkinter.

Hardware requirements include a functional PC or laptop, a webcam with driver installed, and a flashlight or LED for low-light circumstances.

• As previously noted, the Python programming language is employed.
• The code is written using the VS Code.
• This little project is run and created using the Linux Mint operating system.
• An Asus Tuf F17 laptop is utilized.Features include a Core i7 dual core processor, 512GB SSD, 16GB RAM, and an integrated webcam.
• Use a terminal to run the code.
• Tested on Linux Mint, Windows 7, and Windows 11.

V. RESULT AND DISCUSSION

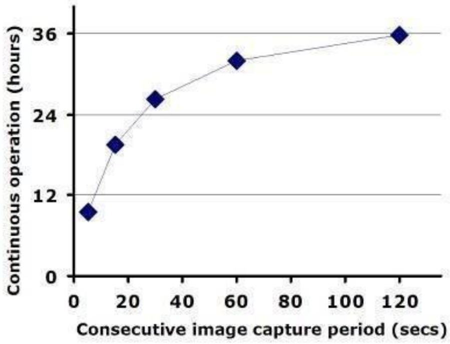

Fig.4. Image capture period v/s time (in hours)

Figure 4 shows the graph of image capture period versus time (in hours), represented by the x and y axes, respectively. Plotting the graph produces a distinct curve that shows the relationship between the two variables.

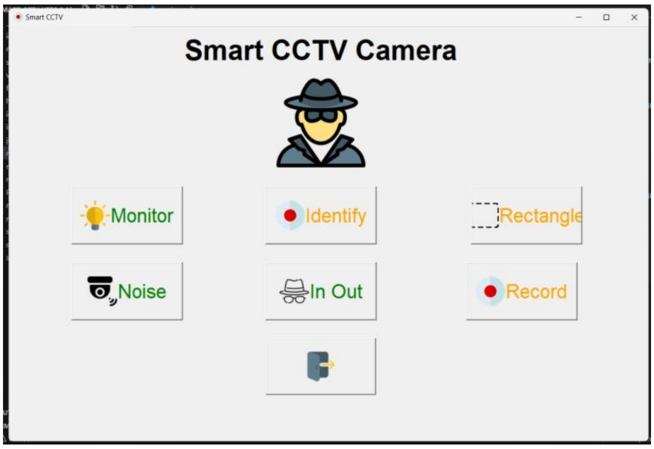

Fig.5. Graphical User Interface

Figure 5 shows the project's graphical user interface. This screen will appear first when we launch the application. The graphic shows the application's title as well as six other icons. Each icon performs a distinct duty, as shown by the name beside it.

Fig.6. Object Detection

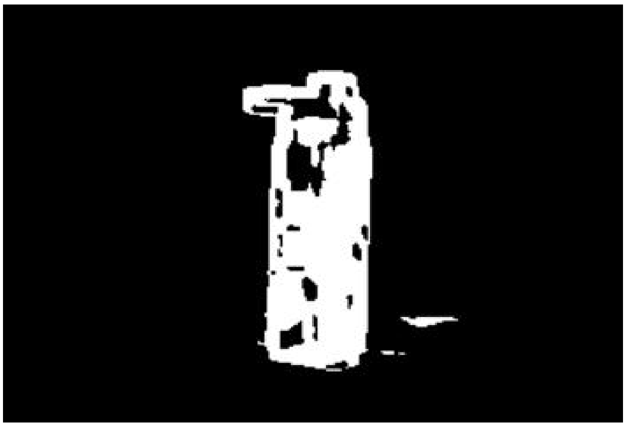

Fig.7. Grayscale Image

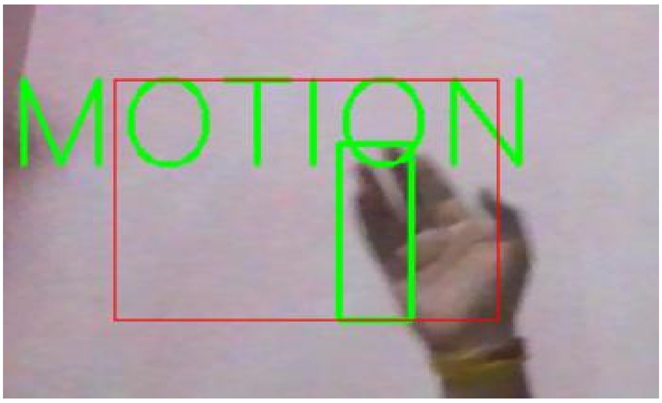

Fig.8. Region (Motion Detected)

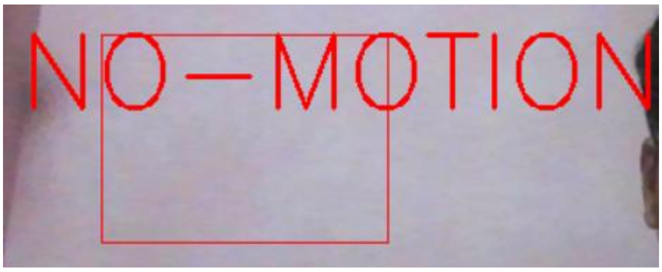

Fig.9. Region based monitoring

The monitor feature will watch the camera's frame. The rectangle button lets us select a frame of reference within the original frame. This functionality can be beneficial for detecting misplaced or stolen goods that were originally in the frame we picked. The in-out feature takes photos of people entering and exiting the frame. These photographs are saved in a folder on our desktop device or computer.

Fig.10. Model Training Window

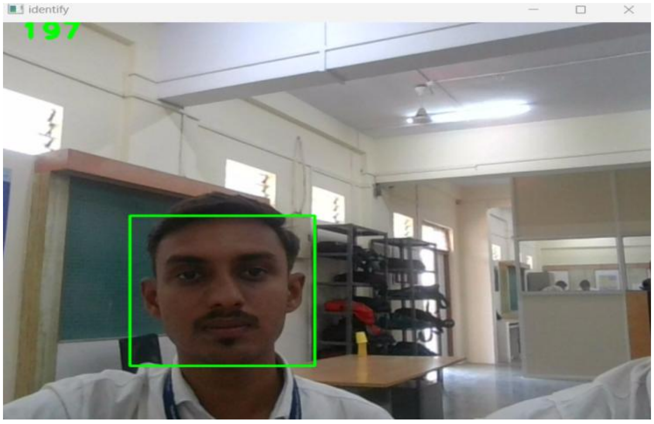

Fig.9. Face Detection model training process

VI. Conclusion

This initiative will significantly impact the future of CCTV and surveillance systems. It will be a stand-alone program that will provide and mix elements from various designs and technologies in a single software application. The benefits of this CCTV system can be summarized into three major points:

1.Offers extra functions such as noise detection, facial recognition, visitor counting, and others that are not readily available in existing surveillance systems.

2.Enables easy access to recorded footage and captured photographs by organizing them into named folders on the device.

3.Identifies misplaced or stolen things in camera frames. Additionally, the face recognition feature helps identify suspects.

VII. FUTURE SCOPE

Looking ahead, the vending machine project presents numerous opportunities for future expansion and improvement. One avenue for advancement lies in enhancing the user interface, potentially integrating a touchscreen display to offer intuitive interaction and detailed product information. Additionally, the integration of advanced payment systems, such as contactless payment methods or mobile payment apps, could provide users with more convenient and secure payment options. Improving inventory management through the implementation of automated tracking systems would enable real-time monitoring of product levels, facilitating timely restocking and preventing out-of-stock situations. Furthermore, the project could benefit from the utilization of data analytics to analyze user preferences and consumption patterns, informing inventory management decisions and marketing strategies. By exploring these and other possibilities, the vending machine project can evolve into a more sophisticated and user-friendly system, delivering enhanced functionality and customer satisfaction.

VIII. References

[1] Bhatti, Muhammad Tahir, Muhammad Gufran Khan, Masood Aslam, and Muhammad Junaid Fiaz. "Weapon detection in real-time cctv videos using deep learning." Ieee Access 9 (2021): 34366-34382.

[2] Shirsat, Samit, Aakash Naik, Darshan Tamse, Jaysingh Yadav, Pratiksha Shetgaonkar, and Shailendra Aswale. "Proposed system for criminal detection and recognition on CCTV data using cloud and machine learning." In 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), pp. 1-6. IEEE, 2019.

[3] Pillai, Manu S., Gopal Chaudhary, Manju Khari, and Rubén González Crespo. "Real-time image enhancement for an automatic automobile accident detection through CCTV using deep learning." Soft Computing 25, no. 18 (2021): 11929-11940.

[4] Arunnehru, J. "Deep learning-based real-world object detection and improved anomaly detection for surveillance videos." Materials Today: Proceedings 80 (2023): 2911-2916.

[5] Dimou, Anastasios, Paschalina Medentzidou, F. Alvarez Garcia, and Petros Daras. "Multi-target detection in CCTV footage for tracking applications using deep learning techniques." In 2016 IEEE international conference on image processing (ICIP), pp. 928-932. IEEE, 2016.

[6] Seng, Kah Phooi, Li-Minn Ang, Leigh M. Schmidtke, and Suzy Y. Rogiers. "Computer vision and machine learning for viticulture technology." *IEEE Access* 6 (2018): 67494-67510.

[7] Gelana, Fraol, and Arvind Yadav. "Firearm detection from surveillance cameras using image processing and machine learning techniques." In *Smart Innovations in Communication and Computational Sciences: Proceedings of ICSICCS-2018*, pp. 25-34. Springer Singapore, 2019.

[8] Maitre, Julien, Kévin Bouchard, and L. Paul Bédard. "Mineral grains recognition using computer vision and machine learning." *Computers & Geosciences* 130 (2019): 84-93.

[9] Holm, Elizabeth A., Ryan Cohn, Nan Gao, Andrew R. Kitahara, Thomas P. Matson, Bo Lei, and Srujana Rao Yarasi. "Overview: Computer vision and machine learning for microstructural characterization and analysis." *Metallurgical and Materials Transactions A* 51 (2020): 5985-5999.

[10] Shah, Juhi, Mahavir Chandaliya, Harsh Bhuta, and Pratik Kanani. "Social distancing detection using computer vision." In *2021 5th International conference on computing methodologies and communication (ICCMC)*, pp. 1359-1365. IEEE, 2021.