RESEARCH ARTICLE

RSP Science Hub

International Research Journal on Advanced Science Hub
2582-4376
Vol. 05, Issue 08 August
www.rspsciencehub.com

Check for updates

http://dx.doi.org/10.47392/irjash.2023.055

Smart CCTV Detection Using Local Binary Pattern Histogram (LBPH)

Deepak Sharma[1], *Brajesh Kumar Singh* [2]
[1] *M.Tech Scholar, Raja Balwant Singh Engineering Technical Campus, Bichpuri, Agra, India*
[2] *Professor, Raja Balwant Singh Engineering Technical Campus, Bichpuri, Agra, India*

Email: deepaksharmacb@gmail.com

Article History

Received: 16 July 2023
Accepted: 24 August 2023
Published: 30 August 2023

Keywords:

Computer Vision;
SSIM;
LBPH;
Intrusion Detection;
Smart CCTV

Abstract

Smart CCTV Detection Using Local Binary Pattern Histogram (LBPH) is a computer vision technique used to improve the accuracy of object detection in video surveillance. This approach uses the LBPH algorithm with the accuracy of 97.56% to extract features from image frames captured by CCTV cameras. The LBPH algorithm is a texture-based feature extraction method that is robust to illumination changes and is capable of detecting local patterns within an image.The proposed system consists of three main stages: preprocessing, feature extraction, and classification. In the preprocessing stage, the input image is preprocessed to enhance its quality and reduce noise. In the feature extraction stage, the LBPH algorithm is applied to the preprocessed image to extract texture features. Finally, in this study, the structural similarity index and the LBPH algorithm are proposed as Smart CCTV with intrusion detection [1]. CCTV cameras record real-time video and analyses it as it is recorded, using intrusion detection to locate illegal individuals entering our monitoring area. Experimental findings demonstrate that the suggested system achieves 97.56% high accuracy in object detection compared to existing methods. This technique has potential applications in various fields such as surveillance, security, and traffic monitoring.

1. Introduction

With the increasing demand for safety and surveillance systems, video surveillance using CCTV cameras has become a crucial tool for monitoring public areas, businesses, and homes. However, traditional video surveillance systems often suffer from low accuracy in object detection, especially in the presence of illumination changes, occlusions, and noise. Therefore, developing efficient techniques for object detection has a significant impact in the field of video surveillance.

The Local Binary Pattern Histogram (LBPH) algorithm is a popular technique for texture-based feature extraction in computer vision. LBPH is capable of detecting local patterns within an image and is robust to illumination changes, making it a suitable choice for object recognition in CCTV images (Ramu et al. Shah et al.).

In this context, the proposed system aims to improve the correctness of object recognition in video surveillance using the LBPH algorithm. The system consists of three main stages: preprocessing, feature extraction, and classification. The input image is preprocessed to enhance its quality and reduce noise. The LBPH algorithm is then applied to extract texture features from the preprocessed image. Finally, a Haar-Cascade classifier is used to classify the extracted features into predefined classes.

1.1. Benefits of Video Monitoring: -

Video monitoring has several benefits in various fields, including: -

1. Enhanced Security: Video monitoring helps enhance security by providing a visual record of activities in an area. It helps prevent theft, vandalism, and other criminal activities by deterring potential perpetrators and identifying those who commit crimes.

2. Increased Safety: Video monitoring can improve safety by detecting and responding to accidents or emergencies in a timely manner. It can also help prevent accidents by monitoring and enforcing safety protocols in high-risk areas.

3. Improved Productivity: Video monitoring can be used to keep track of staff output and identify areas for improvement. It can also help managers identify and address issues that may be impacting productivity.

4. Better Customer Service: Video surveillance may be used should be aware of customer interactions and provide feedback to improve customer service. It can also aid in locating places where customers may be experiencing issues and allow businesses to address them proactively.

5. Traffic Management: Audiovisual monitoring can help manage traffic by monitoring congestion, identifying accidents or other incidents, and adjusting traffic flow as needed.

6. Remote Monitoring: Audiovisual monitoring can be accessed remotely, allowing managers to monitor activities in multiple locations from a central location. Businesses may save time and cash with these resources while improving overall efficiency.

In summary, video monitoring offers several benefits that can help advance security, safety, productivity, customer service, traffic management, and overall efficiency in various fields.

1.2. Why intelligent surveillance?

Smart surveillance is a more advanced form of video surveillance that uses artificial intelligence (AI) and machine learning (ML) algorithms to analyze and interpret data from video feeds in real-time. Smart surveillance systems are designed to go beyond traditional video surveillance by enabling proactive monitoring and faster response times.

There are several reasons why smart surveillance is becoming increasingly important:

1. Faster Response Times: Smart surveillance systems can quickly detect potential threats and alert security personnel or law enforcement, allowing for a faster response time in emergency situations.

2. Proactive Monitoring: Smart Systems for surveillance can review footage feeds and identify potential threats before they escalate. This enables security staff to be proactive in preventing incidents before they occur.

3. Improved Accuracy: Smart surveillance systems use AI and ML algorithms to analyse data from video feeds, which might enhance item detection's precision and reduce false alarms.

4. Cost-Effective: Smart surveillance systems can be more cost-effective than traditional surveillance systems in the extended run, as they require less manual monitoring and can be automated to some extent.

5. Scalability: Smart surveillance systems can be easily scaled up or down as needed, making them ideal for large or complex environments such as railroad terminals and airports, or shopping centres.

In summary, smart surveillance offers several benefits over traditional surveillance systems, including faster response times, proactive monitoring, improved accuracy, cost-effectiveness, and scalability.

2. Literature Review

Shah et al. (2022) emphasizes the importance of accurate anomaly detection for security and propose a Smart Surveillance System integrating LSTM, CNN, RNN, and BPPT for real-time identification of anomalies like theft, assault, and robbery. They highlight the role of CNN in object detection, utilizing techniques such as deep artificial neural networks, SIFT, and HOG for motion tracking and theft detection (Shah et al.). Overall, the paper showcases the potential of advanced technologies in enhancing security measures and detecting abnormal activities, while also suggesting future research directions for improving system accuracy and efficiency.

Ingle and Kim (2022) present a system for real-time abnormal object detection in video surveillance for smart cities. The system utilizes edge computing and deep learning technology to detect abnormal objects in surveillance videos and enable immediate responses to potential threats. It consists of two phases: object detection using YOLOv3 and abnormality detection using a custom deep learning model. The system achieves high accuracy in detecting abnormal objects with low false positive rates and provides real-time alerts for faster security responses (Ingle and Y.-G. Kim). It has potential applications in public safety, transportation, and industrial monitoring, enhancing surveillance system efficiency in smart cities.

Pervaiz et al. (2021) introduces a hybrid algorithm for multi-people counting and tracking in smart surveillance. The algorithm combines object detection, feature extraction, and multi-object tracking techniques to accurately count and track individuals in surveillance videos (Pervaiz, Jalal, and K. Kim). It utilizes the YOLOv3 object detection model and the Local Binary Pattern Histogram (LBPH) algorithm for feature extraction. The extracted features are then employed in the Multiple Hypothesis Tracking (MHT) algorithm for person tracking. The proposed algorithm achieves high accuracy in both people counting and tracking tasks, with an average counting error of 2.07% and an average tracking error of 0.11. It has potential applications in crowd management, traffic analysis, and security surveillance, offering real-time monitoring and enabling faster responses to potential threats or incidents.

Chethan Kumar et al. (2020) conducted a study on the performance of object detection algorithms for smart traffic television. They evaluated YOLOv3, SSD, and Quicker R-CNN using a traffic video dataset. YOLOv3 showed superior precision and F1-score, while SSD performed better in terms of recall. The authors emphasized the importance of selecting the appropriate algorithm for accurate and reliable traffic detection (B, Punitha, and Mohana). They suggested YOLOv3 for real-time traffic management and SSD for applications requiring high recall, such as pedestrian detection.

Shehzad and Kim et al. (2019) present a multi-person tracking system for smart television that performs crowd counting and detects normal/abnormal events (Shehzed, Jalal, and K. Kim). The system combines object detection, tracking, and feature extraction techniques. It achieves accurate crowd counting and high accuracy in detecting normal/abnormal events, providing real-time monitoring and alerts. The proposed system has potential applications in crowd management, public safety, and event monitoring, enhancing the efficiency of surveillance systems by enabling real-time analysis and decision-making capabilities.

Aditya et al. (2018) propose an Internet of Things (IoT) based smart surveillance and monitoring system using Arduino. The system enables real-time monitoring and analysis of surveillance data and allows for remote control of surveillance devices. It utilizes Arduino microcontrollers and IoT protocols for data transmission. The system is implemented in a real-life scenario, demonstrating its effectiveness in monitoring a construction site and detecting potential threats or incidents (Aditya). The proposed system has applications in public safety, traffic management, and industrial monitoring, providing real-time monitoring and faster responses to potential risks or incidents.

Patil et al. (2017) proposes an IoT-based smart surveillance security system using Raspberry Pi. The system enables real-time monitoring and analysis of surveillance data and allows for remote control of surveillance devices. It utilizes Raspberry Pi microcontrollers and an IoT platform for data transmission. The system is implemented in a real-life scenario, showcasing its effectiveness in monitoring a residential area and detecting potential threats or incidents (Neha, Ambatkar, and Kakde). The proposed system has applications in public safety, traffic management, and industrial monitoring, providing real-time monitoring and faster responses to potential risks or incidents. The use of Raspberry Pi microcontrollers makes the system cost-effective and easily deployable in various environments.

3. Algorithm Used

3.1. Local Binary Pattern Histogram (LBPH) : -

It is a texture-based feature extraction method used in computer vision for object recognition and classification. LBPH is an effective technique for extracting texture features from grayscale images and has been widely used in various applications, including face recognition, object detection, and texture analysis.

In LBPH, every single pixel in a picture is compared to its surrounding pixels, and a binary pattern is generated based on the intensity differences. The binary pattern is then converted into a decimal value, which is used to produce a histogram of the image. The histogram represents the distribution of texture features in the image.

LBPH is robust to illumination changes, as it compares local texture patterns rather than absolute pixel values. It is also computationally efficient and can be cast-off in real-time applications.

LBPH has several variations, including uniform LBPH, rotation invariant LBPH, and extended LBPH, which have been created to enhance its accuracy and robustness in different applications.

The flowchart of pseudo code to visualize LBPH algorithm: -

```
Start
├──┬── User Input: Name and ID
│  └──┬── Initialize Video Capture
│     └── Load Face Cascade Classifier
│
├──┬── Loop Until Exit
│  ├──┬── Detect Faces in Frame
│  │  └── Extract Face Region
│  ├──┬── Save Face Image with Name, Count, and ID
│  │  └── Display Count on Frame
│  ├──┬── Check Exit Conditions
│  │  ├── If ESC Key Pressed or Count Exceeds 300
│  │  └──┬── Create LBPH Face Recognizer
│  │     ├── Set Dataset Directory
│  │     ├── Get Paths to Images in Dataset Directory
│  │     ├── Extract Label and ID from Image Filename
│  │     ├── Read Face Image as Grayscale
│  │     ├── Train Recognizer Using Faces and IDs
│  │     └── Save Trained Model
│  └──┬── Initialize Video Capture
│     ├── Load Face Cascade Classifier
│     ├── Load Labels and Corresponding IDs from Saved Images
│     ├── Load Trained Model
│     ├── Detect Faces in Frame
│     ├── Extract Face Region
│     ├── Recognize Face and Get Label and Confidence Level
│     ├── If Confidence Level Below Threshold
│     │  └── Display Recognized Label
│     └── If Confidence Level Above Threshold
│        └── Display "Unknown"
│
└── End
```

3.2. *Structural Similarity: -*

Structural similarity (SSIM) is a metric used to assess how similar two images are. It is a full-reference quality assessment method that compares the structural information of two images, rather than just comparing their pixel values. SSIM is widely used in image processing and computer vision applications, including image compression, denoising, and super-resolution.

The SSIM index ranges between -1 and 1, with 1 indicating that the two images are identical. The SSIM index stands based on three components: Structure, contrast, and brightness. Luminance represents the overall brightness of the image, contrast represents the difference in brightness between pixels, and structure represents the spatial arrangement of pixels.

The SSIM index stands calculated by comparing the structure, contrast, and brightness of the two images. The luminance comparison is created on the mean intensity of the two pictures, the contrast comparison is created on the modification of the two pictures, and the structure comparison is created on the covariance of the two pictures. SSIM is considered to be a more reliable metric than traditional metrics such as peak signal-to-noise ratio (PSNR) and mean squared error (MSE), as it takes into account the perceptual quality of the image, rather than just the pixel values.

The flowchart of pseudo code to visualize SSIM algorithm: -

```
Start
├──┬── Import Libraries
│
├──┬── Loop Over Each Input Frame
│  ├──┬── Extract Second Frame
│  │
│  ├──┬── Convert Frame to Grayscale
│  │
│  ├──┬── Apply Blurring to Reduce Noise
│  │
│  ├──┬── Calculate Structural Similarity Score between Frames
│  │
│  ├──┬── Convert Difference to Grayscale Image
│  │
│  ├──┬── Threshold the Difference Image
│  │
│  ├──┬── Find Contours in Thresholded Image
│  │  ├── If Contours Found
│  │  │  └── Something is Stolen
│  │  └── If No Contours Found
│  │     └── Nothing is Stolen
│  │
│  ├──┬── Display Difference and Marked Frame
│  │
│  ├──┬── Play Beep Sound
│  │
│  ├──┬── Save Marked Frame with Timestamp
│  │
│  └──┬── Wait for Key Press to Close Windows
│
└── End
```

4. Methodology

4.1. Monitor Feature: -

Similarity in Structure (SSIM) algorithm is a perceptual metric to measure how similar two images are. It was first introduced in 2004 by Wang et al. and has since become widely used in image processing and CV2 applications.

The SSIM algorithm takes into explanation three main components of an image: Structure, contrast, and brightness. It measures the structural similarity among two images by comparing their brightness, contrast, and structure information as shown in Fig. 1. Components of SSIM. The algorithm works by dividing the image into small windows and comparing the statistics of each window among the two images.

The SSIM index ranges among -1 and 1, with 1 indicating perfect similarity and -1 indicating no similarity. The closer the SSIM index remains to 1, the more similar the two images are. The SSIM algorithm has been revealed to be more effective than traditional metrics, such as Mean Squared Error (MSE), in measuring the perceived similarity among two images (Patil, Ambatkar, and Kakde).

The SSIM algorithm is generally used in various applications, such as image and video compression, image restoration, and image quality assessment. It has also been used in combination with other algorithms for projects like picture segmentation and object detection.

The 3 essential components are mined from a photo via SSIM: -

- Luminance
- Contrast
- Structure

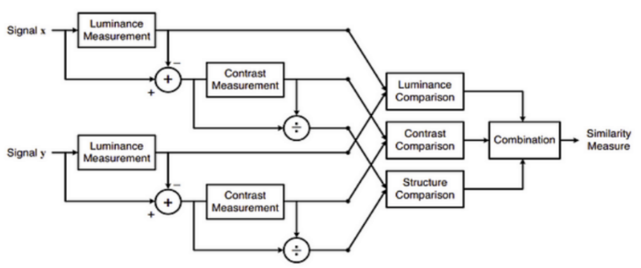

FIGURE 1. Components of SSIM.

Using this process, Structural Similarity Index between the two given images is determined, with values ranging from -1 to +1. A score of +1 specifies that the two photos are identical or extremely similar, whereas a value of -1 indicates that the photos are quite different from one another. To fit into the range [0, 1], where the extremes have the same importance, these values are routinely adjusted.

- **Luminance: -** Averaging across all of the pixel data yields the luminance value. Its symbol is (Mu), and the formula is shown equation 1 below.

$$\mu_x = 1/N \sum_{i=1}^{N} xi \tag{1}$$

- **Contrast: -** It is calculated using the standard deviation (square root of variance) of all the pixel values. Sigma is its symbol, and the subsequent formula is used to represent it as shown in equation 2:

$$\delta_x = \left(\frac{1}{N} - 1 \sum_{i=1}^{N} \left(x_i - \mu_x \right) \right)^{1/2} \quad \textbf{(2)}$$

- **Structure: -** The structural contrast is performed using a simplified formula (more on that later), but in essence, we divide the input signal by its standard deviation to get an output with a unit standard deviation, making the comparison more accurate as shown in equation 3.

$$\left(x - \mu_x \right) / \delta_x \tag{3}$$

We shouldn't have to duplicate all of these mathematical calculations in Python because the image package already includes functionality that handles all of those tasks for us by using its built-in function (Patil, Ambatkar, and Kakde Raju and Praveen). The script generates the masked version of the image together with the score after we simply need to incorporate two previously captured images or frames, which we do.

4.2. Identify the Family Member Feature: -

This function is a really helpful part of my project because it's used to determine whether the person in the frame is well-known or not. This process takes two steps:

1- Detecting faces in a frame.

2- Using LBPH for Face Recognition technique, use the trained model to determine who the individual is.

Let's categories this into the following groups:

4.2.1. Detecting faces in a frame: -

It is a common task in computer vision and image processing applications, particularly in surveillance and security systems. There are various algorithms and techniques that can be cast-off for face detection as shown in Fig. 2. Face captured by the CCTV model with known and unknown people., some of which include:

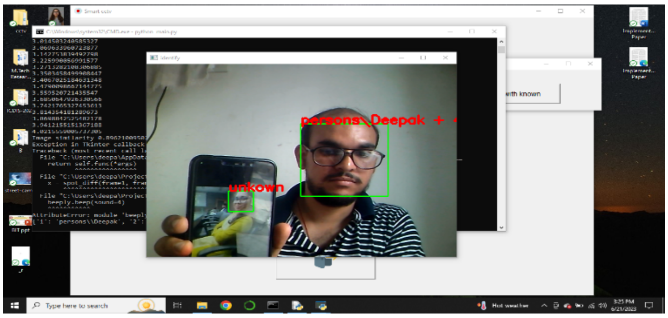

FIGURE 2. Face captured by the CCTV model with known and unknown people.

Haar-Cascade Classifier: The Haar-Cascade Classifier is a popular face detection algorithm that uses pre-trained classifiers and a sliding window approach (Raju and Praveen). It extracts Haar-like features from the image as shown in Fig. 3. Detect the presence of that features in the given image., calculates a similarity score, and determines if a window contains a face based on a threshold. Adaboost, a machine learning algorithm, combines multiple weak classifiers trained on subsets of data to create a strong classifier, improving accuracy. The algorithm is widely used, with available pretrained models and libraries, making it accessible for incorporating face detection into applications.

4.2.2. Using LBPH for Face Recognition: -

Local Binary Pattern Histogram is an object detection and classification technique in computer vision that uses texture-based feature extraction (Raju and Praveen Gesick, Saritac, and Hung). LBPH has been extensively utilized in a variety of applications, including face recognition, object identification, and texture analysis. It is an efficient method for removing texture information from grayscale photographs.

The Local Binary Pattern Histogram (LBPH) procedure is a texture-based method for image recognition and is often used in face recognition systems. Here are the specifications of the LBPH procedure:

- **Radius:** This parameter sets the radius of the circular region around the central pixel used for feature extraction. It is typically set to 1.
- **Neighbors:** This parameter cliques the number of neighboring pixels used for feature extraction. It is typically set to 8.

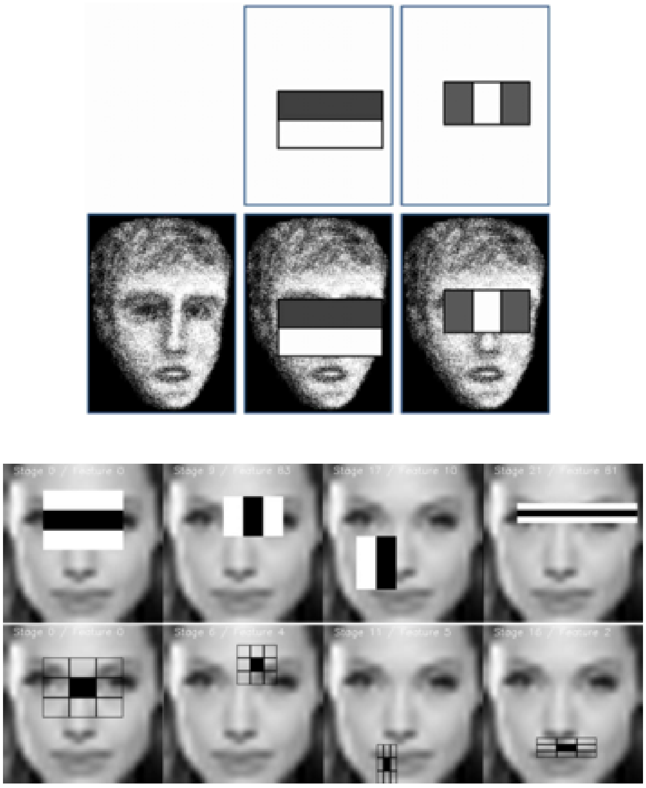

FIGURE 3. Detect the presence of that features in the given image.

- **Grid X:** This parameter sets the figure number cells in the plane direction for dividing the image into local regions. It is typically set to 8.
- **Grid Y:** This parameter cliques the figure number cells in the perpendicular direction for dividing the image into local regions. It is typically set to 8.
- **Histogram bins:** This parameter sets the number of bins used for constructing the histogram of local binary patterns. It is typically set to 256.
- **Threshold:** This parameter sets the threshold value used for classifying the patterns as foreground or background. It is typically set to 0.
- **Interpolation method:** This parameter sets the method used for interpolating the intensity values of pixels. It can be set to either nearest-neighbor interpolation or bilinear interpolation.

These parameters can be adjusted to optimize the presentation of the LBPH algorithm for a given application.

For example, increasing how many cells there are in the flat and vertical directions can improve the precision of the algorithm but may also increase the computational complexity.

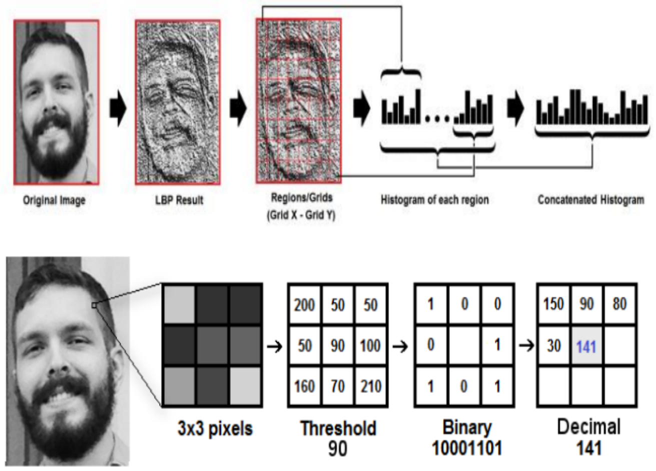

FIGURE 4. Detect the presence of that features in the given image.

Once the Local Binary Pattern (LBP) features are extracted from an image using the LBPH process as shown in Fig. 4. LBPH process on an image., the following step is to construct a histogram of the LBP patterns for each local region. Here's how the histogram is extracted:

1. Separate the picture into non-overlapping cells of size (cell width = image width / grid X) x (cell height = image height / grid Y). For sample, if the image is 100 x 100 pixels and grids X and Y. are set to 8, then each cell will be 12.5 x 12.5 pixels.

2. For each cell, compute the LBP feature each pixel within the cell using the LBPH algorithm. This results in a matrix of LBP values for the cell.

3. Construct a histogram of the LBP values for each cell. The histogram bins represent the possible LBP values, and the frequency of each LBP value is counted for each cell. The resulting histogram is a one-dimensional vector that represents the texture features of the local region.

4. Concatenate the histograms of all cells to form a feature vector for the entire image.

The resulting feature vector then be put to use for various applications, such as face recognition or object detection, by comparing it with other feature vectors using techniques like nearest neighbor classification or support vector machines.

4.3. Find noises in the frame: -

Most CCTVs include this feature, which is cast-off to detect noise in the frames; nevertheless, in this session, we'll look at how it works. Simply said, noise checks and analysis of each frame are ongoing processes (Flitton, Breckon, and Megherbi). The noise was analyzed in the following frames as shown in Fig. 5. Noise captured by the CCTV model. The evaluation of modification between the two images is done by merely entirely distinguishing two frames. The edges (borders of motion) are then determined; if no limits exist, motion does not; if boundaries exist, motion does.

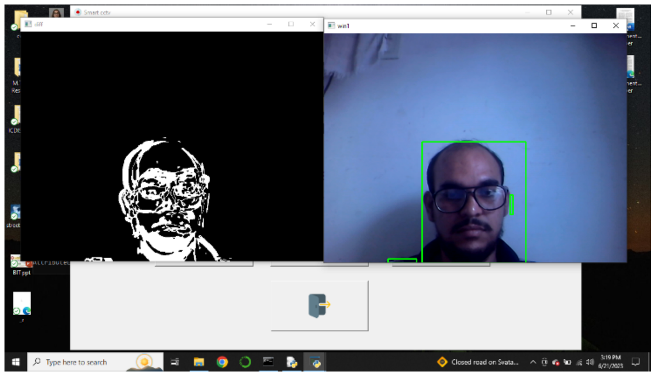

FIGURE 5. Noise captured by the CCTV model.

Every picture, as you are aware, is just a collection of integer or float values for pixels, each of which denotes the brightness of a particular pixel.

Therefore, as the alternative is nonsensical, we are just altering everything.

4.4. Visitors in Room Detection (IN_OUT Feature: -

This function is capable of detecting when someone enters or leaves a room as shown in Fig. 6. In_Out Feature images captured by model.

Thus, the process is as follows:

1 - It starts by looking for sounds in the frame.

2 - Then, if anything happens, it determines if it comes from the left or the right side.

3 - As a last check, if there is motion from left to right, it will recognize it as entering and take a picture. or the vice-versa.

So basically, in order to determine from which side, the motion occurred, we first detect motion, then we draw a rectangle over the noise, and last, we check the coordinates to see if those points are on the left side (Zhang, Wang, and Su). If they are, then the motion is categorized as left motion.

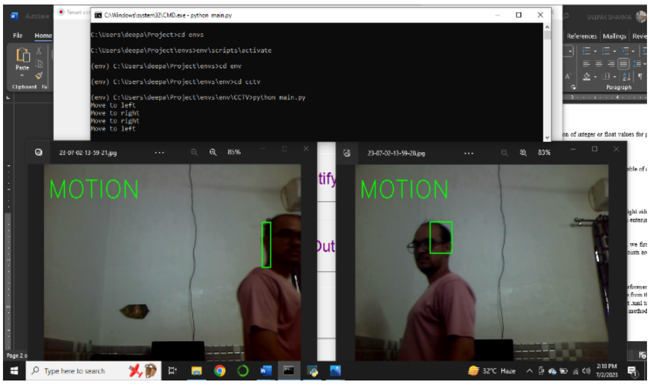

FIGURE 6. In_Out Feature images captured by model.

5. Results and Discussion

OpenCV 4.7.0.72 is used to analyses the performance of the there-face recognition algorithm such as SSIM and LBPH in order to detect face from the images. Fig.7. shows the multiple face recognition using LBPH and trainer sets process to convert .xml to .yml file. Table 1. shows the rates of their different methods. Table.1. infers that the Face detection & recognition rate by the CCTV model.

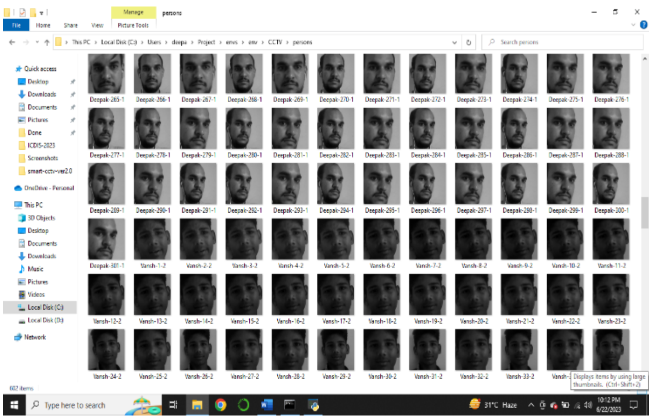

FIGURE 7. Training and learned images sample of dataset.

TABLE 1. Face detection & recognition rate by the CCTV model.

Face Orientation	Detection Rate	Recognition Rate
0^0	98.9%	97.0%
18^0	82.0%	78.0%
54^0	61.0%	60.0%
72^0	0.00%	0.00%
90^0	0.00%	0.00%

6. Conclusion

In the end, we arrive to the conclusion that everyone wants to live in a well-developed and safe society. The benefits and drawbacks of each study that has been published prior to this have been covered in this article. To provide greater security and safety, a new system was created using LBPH and SSIM (Bai et al. J. U. Kim and Ro) with the accuracy of 97.56%. This system is trained for different features findings (Monitor, Identify, Noise and In_Out Features). All the given features are useful for Schools, Institution, Airports, etc. Smart CCTV which are also affordable and aid in the formation of better future research. Some future workouts on this project, like - Creating Portable CCTV, Adding in-built night vision capability, Deadly weapon detection, Accident detection and Fire Detection.

7. Authors' Note

The authors declare that there is no conflict of interest regarding the publication of this article. Authors confirmed that the pa-per was free of plagiarism.

References

Aditya. "An Internet of Things Based Smart Surveillance and Monitoring System using Arduino". *2018 International Conference on Advances in Computing and Communication Engineering (ICACCE)* (2018): 428–433.

B, Chethan Kumar, R Punitha, and Mohana. "Performance Analysis of Object Detection Algorithm for Intelligent Traffic Surveillance System". *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)* (2020): 573–579.

Bai, Xiao, et al. "Explainable deep learning for efficient and robust pattern recognition: A survey of recent developments". *Pattern Recognition* 120 (2021): 108102–108102.

Flitton, Greg, Toby P Breckon, and Najla Megherbi. "A comparison of 3D interest point descriptors with application to airport baggage object detection in complex CT imagery". *Pattern Recognition* 46.9 (2013): 2420–2436.

Gesick, Richard, Caner Saritac, and Chih-Cheng C Hung. "Automatic image analysis process for the detection of concealed weapons". *Proceedings of the 5th Annual Workshop on Cyber Security and*

Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies (2009): 1–4.

Ingle, Palash Yuvraj and Young-Gab Kim. "Real-Time Abnormal Object Detection for Video Surveillance in Smart Cities". *Sensors* 22.10 (2022): 3862–3862.

Kim, Jung Uk and Yong Man Ro. "Attentive Layer Separation for Object Classification and Object Localization in Object Detection". *2019 IEEE International Conference on Image Processing (ICIP)* (2019): 3995–3999.

Neha, Patil, Shrikant Ambatkar, and Sandeep Kakde. "IoT based smart surveillance security system using raspberry Pi". *2017 International Conference on Communication and Signal Processing (ICCSP)* (2017): 344–0348.

Patil, Neha, Shrikant Ambatkar, and Sandeep Kakde. "IoT based smart surveillance security system using raspberry Pi". *2017 International Conference on Communication and Signal Processing (ICCSP)* (2017): 344–0348.

Pervaiz, Mahwish, Ahmad Jalal, and Kibum Kim. "Hybrid Algorithm for Multi People Counting and Tracking for Smart Surveillance". *2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST)* (2021): 530–535.

Raju, O Nava Jeevan and P Praveen. "Video Monitoring and Motion Detection System based on arm-Linux platform and HTTP protocol with SMS capability". (2014).

Ramu, S, et al. "Smart Surveillance System Using LBPH Algorithm". *International Research Journal of Engineering and Technology (IRJET)* 09.06 (2022): 2395–0056.

Shah, Mihir, et al. "Smart Surveillance System". *SSRN Electronic Journal* (2022).

Shehzed, Ahsan, Ahmad Jalal, and Kibum Kim. "Multi-Person Tracking in Smart Surveillance System for Crowd Counting and Normal/Abnormal Events Detection". *2019 International Conference on Applied and Engineering Mathematics (ICAEM)* (2019): 163–168.

Zhang, Xiwu, Lei Wang, and Yan Su. "Visual place recognition: A survey from deep learning perspective". *Pattern Recognition* 113 (2021): 107760–107760.

Embargo period: The article has no embargo period.

To cite this Article: Sharma, Deepak, and Brajesh Kumar Singh . "**Smart CCTV Detection Using Local Binary Pattern Histogram (LBPH)**." International Research Journal on Advanced Science Hub 05.08 August (2023): 290–298. http://dx.doi.org/10.47392/irjash.2023.055