Khulna University Studies Special Issue ICSTEM4IR: 19-32, 2022

ORIGINAL ARTICLE

SCIENCE & ENGINEERING

DESIGN AND DEVELOPMENT OF A SURVEILLANCE ROBOT

Md Khaled Hasan[1,*], Gazi Salahuddin[2], Sayef Ali Khan[3], and Md. Shamim Ahsan[1]

[1]*Electronics and Communication Engineering Discipline, Khulna University, , Khulna 9208, Bangladesh*
[2]*Echologyx Ltd., Uttara 1230, Bangladesh*
[3]*Information and Communication Technology Friedrich-Alexander-Universität, Erlangen-Nürnberg Erlangen, Germany*

KUS: ICSTEM4IR-22/0167

Manuscript submitted: August 11, 2022 Accepted: September 25, 2022

Abstract

Conventional surveillance which was done by human, is a dull job and prone to many mistakes. Additionally, manpower required for monitoring is expensive and is not suitable for weltering in remote places. Surveillance camera-based monitoring system is a temporary solution in this scenario. However, surveillance cameras are fixed to a certain position with limited coverage area. Furthermore, surveillance cameras defunct during and after natural disasters such as earthquakes, storms, etc., while there is a necessity for search and rescue of dwindling survivors. We propose a surveillance robot that overcomes the surveillance systems' restricted coverage area problem with a light weight and low-cost robot structure, attached with a movable surveillance camera. Instead of recording video footage in passive, the system actively telecast visual information to a Raspberry-Pi system connected with Wi-Fi. A face detection system based on Viola-Jones algorithm is used to detect faces on real time basis. A user-friendly and easy to manage Graphical User Interface (GUI) is introduced using PyQt5. The experimental results show that the robot can detect single to multiple faces either eyes open or closed on real time basis.

Keywords: Word, lower case, word

*Corresponding author: <khaled.kuece15@gmail.com>
DOI: https://doi.org/10.53808/KUS.2022.ICSTEM4IR.0167-se

Introduction

Many different fields like banks, household, elevators, airport, mining accidents, urban disasters, hostage situations explosions et cetra requires surveillance. Traditional human monitoring is accomplished by stationing personnel near vulnerable area and monitoring for changes on a continuous basis. Humans, on the other hand, have their limits and prone to the dullness of surveillance works. With the recent developments of computer hardware, machine vision, and miniature peripheral equipment rapidly, surveillance system has moved from human to computer. Surveillance cameras for their fixed nature have limited capturing range. On the other hand, surveillance cameras attached to a robot have superiority for their flexible mobility. However, these surveillance robots can not operate with minimal human intervention and interpret any visual information. Visual information interpretation, an easy task for human is complicated to perform preeminently on computer.

Abudhagir *et al.* (Abudhagir et al., 2022) proposed an upgraded version of FACENET based on the use of Faster Regional Convolutional Neural Networks (FRCNN) to boost performance. FR-CNN uses a technique called hard negative mining to distinguish between positive and negative samples. The images in the dataset are one shot learned images that provided the best face detection performance. While passing through convolution sheets, the method used triplet loss to prioritize correlated example pictures. Multi-task cascaded convolutional neural network (MTCNN) for heterogeneous face detection was presented by Yang and Zhang (Yang and Zhang, 2022). To achieve rapid and efficient face detection, the system employs a candidate frame with a classifier. Proposal network (P-Net) generates candidate window, which is subsequently filtered for high precision candidate frames and selected by reduced network (R-Net). Finally, the output network (O-Net) generates boundary boxes and face landmarks. Liu *et al.* (Liu et al., 2022) proposed combining deep learning with the video sequence's continuity. To train Rest-Single Shot MultiBox Detector (SSD), the method first uses residual nets as the basic network of the SSD. Then, using a video sequence, Rest-SSD based training is used for face detection. To track consecutive n frames, kernel co-relation filtering is utilized, followed by the weighted average approach to determine the best tracking confidence result, which is then applied for current frame sets. However, the above literature despite being the state of the art best face detection models are very heavy and computationally time consuming. The methods require higher training examples and require higher computational power and heavy computers. Moreover, these models can not be used on lightweight embedded systems which abdicate them from real life implementations.

Ullah *et al.* (Ullah et al., 2022) presented a machine learning and deep learning based real time surveillance system with face detection on closed-circuit television (CCTV) images. The framework employed principal component analysis (PCA) and CNN as feature extractors while used and compared K-nearest neighbor (KNN), decision tree, random forest, and CNN as classifiers. Wibowo *et al.* (Wibowo et al., 2021) suggested an improvement of face detection using CCTV by using RETINAFACE as a post processing step. The CCTV images suffers from illumination variation. Once this illumination variant images are fed to RETINAFACE, the accuracy drops at a significant rate. That's why the method captures images through CCTV and fixes illumination using homomorphic filtering systems and use RETINAFACE as post-processing. Nevertheless, the above literature outlines the conventional surveillance system which implements CCTV and the methods are rigid and has a very limited coverage area. Kanagaraj *et al.* (Kanagaraj et al., 2022)

presented a spy robot with face detection capable of moving to harsh places like mines, dense forests, etc. The framework has been implemented on Raspberry Pi and the robot can be wirelessly controlled by the user using a Wi-Fi technology. However, the system employs rigid camera setup which is incapable of turning around in case of different angle is needed. Moreover, the face detection algorithm used in the system is slow and can not detect more than one face in one frame. Additionally, a major face detection problem such as eye state is not discussed in the study. Salh and Nayef (Salh and Nayef, 2013) proposed a face detection based surveillance robot. The method uses a fusion of PCA and Linear Discriminant Analysis (LDA) for feature extraction. Additionally, Support Vector Machine (SVM) was implemented to enhance detection rate. However, the model is computationally complex and requires high processing. Budiharto (Budiharto, 2015) suggested an intelligent surveillance robot which was implemented using three distance sensors, avoids obstacles in the robot's route. The model utilizes a face detection system based on Neural Network. However, the system is computationally costly and complex. Renuka *et al.* (Renuka et al., 2018) proposed an automatic enemy detection robot based on face detection. The paper uses arduino and android platform in the implementation. Yet the method is limited to Bluetooth connectivity only.

In this paper, we proposed a compact surveillance robot with face detection capabilities. Following are some of the contributions of the paper:

- A light weight and cost effective design of the robot has been presented.
- An user friendly and easy to use GUI has been developed.
- A real time fast face detection system with higher accuracy using light weight Viola-Jones algorithm has been implemented which solves the need of heavy deep learning algorithm implementation for high performance on embedded system.
- The novelty of the study is in combining existing methods of face detection, GUI design, robot design, and surveillance system design into a compact system.

The rest of this paper is organized as follows: In Section 2 we briefly explain the mathematical and background theory behind face detection system. Section 3 gives an overview of the materials and methods utilized in building the surveillance robot. Section 4 provides the experimental results utilizing the intelligent surveillance robot. Finally, Conclusion and future directions are given in Section 5.

Background

Haar like features

We used face detection procedure which is based on features in an image instead of pixel based approach. The main reason behind this is feature based approach is faster and can encode ad-hoc domain knowledge. As a feature based approach, rectangle features are employed. There are several kinds of rectangle features, some of them are epitomized in (Figure 1).

White region of the rectangle represents 1 and dark region depicts 0. A transition from 0 to 1 or vice versa represents a difference $1 - 0 = 1$ or what can be called as an edge. Every rectangle slides over an image like in (Figure 2) and produce difference matrix. For 1st difference matrix, the

value is $0.6 - 0.5 = 0.1$. After sliding the rectangle through the whole image, another matrix just containing the differences are made. The pixel values in a gray image are between 0 and 1 and so the closer the value is towards 1 the brighter the pixel is and darker the nearest of 0. On the other hand, the closer the difference value to 1, there is a larger possibility for an edge.

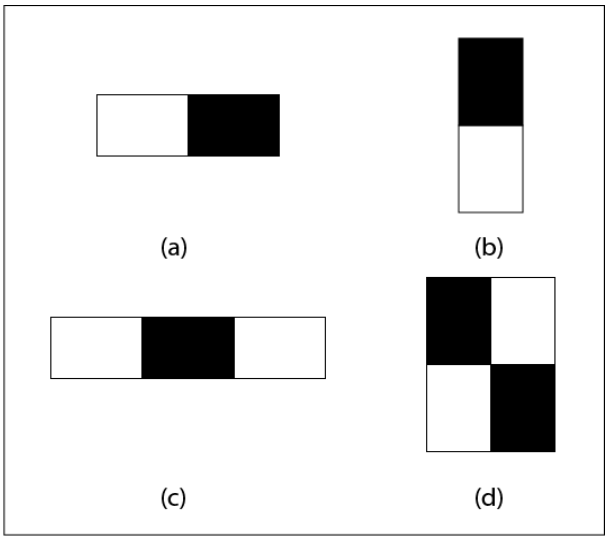

Figure 1: Example of rectangle features. Two rectangle features are shown in (a), (b). Fig. (c) shows a three rectangle feature, and (d) epitomizes a four rectangle feature.

The regions in the rectangle have the same shape and size. If there is 5 pixels in the white region of two rectangle feature then there will be 5 pixels also in the black region. Overall, rectangle shape and size is enlarged or shortened in order to fit a face in the image. So, the overall rectangle can be of 100×100 shape. In this case the subtraction is done from the sum of the pixels in white region to the sum of the pixels in dark region.

Integral Image

There is a huge computation complexity when a large shaped rectangle have to slide over an image. Let's say we want to find out sum of all the pixels in a 6×6 rectangle as shown in (Figure 3). The conventional method is summing up all the pixel values and will require all the 36 array references. This is computationally very expensive because the sliding process needs the summing up lots of times. Using an integral image, the summing up process no matter how large the pixel values are, can be done only with four array references. The integral image as shown in (Figure 3(a)), in any location (x,y) is the summation of pixels on the left and to the above.

Using conventional method the 6×6 rectangle has a summation of 72 pixels which uses all 36 array references. On the other hand, using integral image the value is calculated as shown in (Figure 4(a)) $(4 + 1 - 2 - 3)$. So the sum of the pixels in 6×6 rectangle is $128 + 8 - 32 - 32 = 72$ which is same as the real value.

Using integral image, any rectangle sum can be computed using four array references and two rectangle sum can be calculated using 8 array references as shown in (Figure 4(a)) and (Figure 4(b)). If there are two adjacent rectangles, the sum can be calculated using 6 array references as shown in (Figure 4(c)).

DOI: https://doi.org/10.53808/KUS.2022.ICSTEM4IR.0167-se

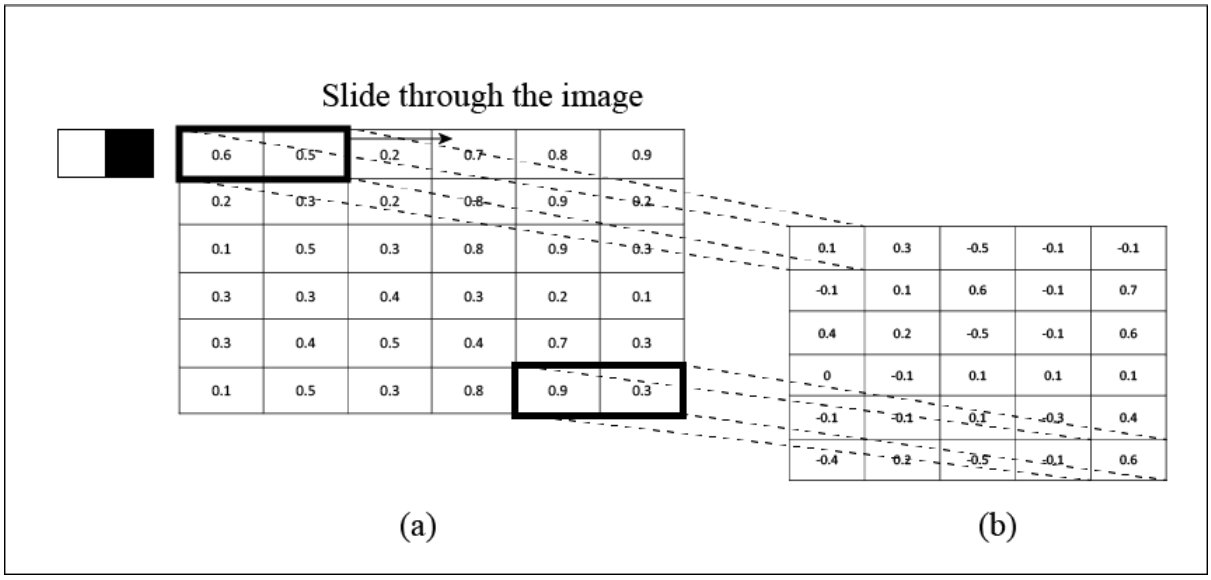

Figure 2: Sliding of rectangle features in an image (a) a gray level image in pixel values (b) difference matrix after sliding a two rectangle feature.

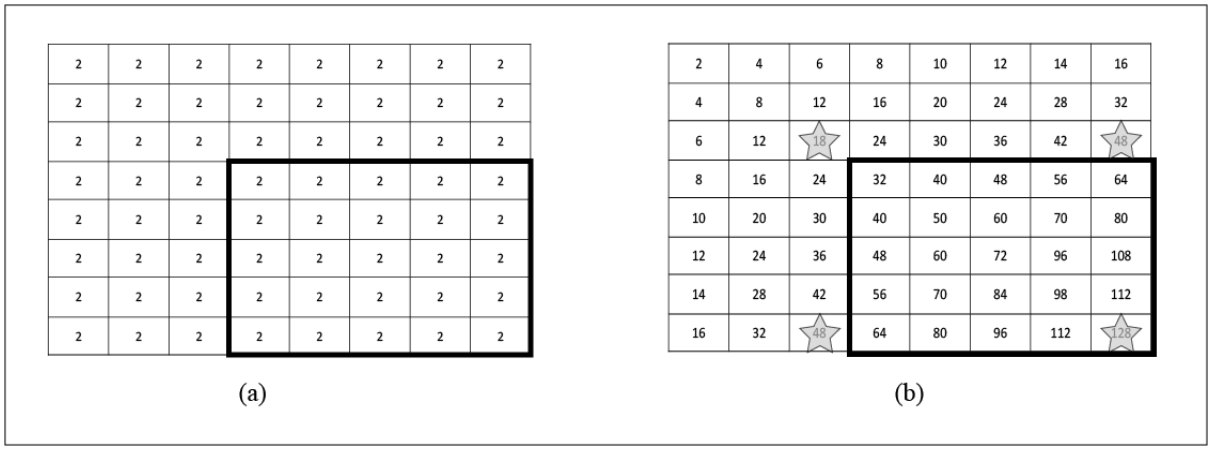

Figure 3: Example of integral image: (a) an 8 × 8 sized input image expressed with pixel values.Using conventional method, the 5 × 5 rectangle has a summation of 50 pixels, which uses all 25 array references; (b) integral image of the input image. Using this integral image, the value is calculated as (4 + 1 – 2 – 3). Here, 1, 2, 3, 4 are the positions of the rectangles shown with circles. So, the sum of the pixels in 5 × 5 rectangle is 128 + 18 – 48 – 48 = 50, which is same as the real value, using only 4 array references instead of 25.

AdaBoost

In a 24×24 frame, the rectangle features are over 180,000 (Viola and Jones, 2001). Yet only a few of the features are important in classifying a face or non-face. AdaBoost algorithm generates those strong classifiers after boosting the weak classifiers. To create strong classifiers, AdaBoost checks performance of all the classifiers. Classifiers are checked on all the sub-regions of an image to measure performance. Some sub-regions will produce better response than others meaning a better probability to contain face. These classifiers will be given higher importance and separated as strong classifiers.

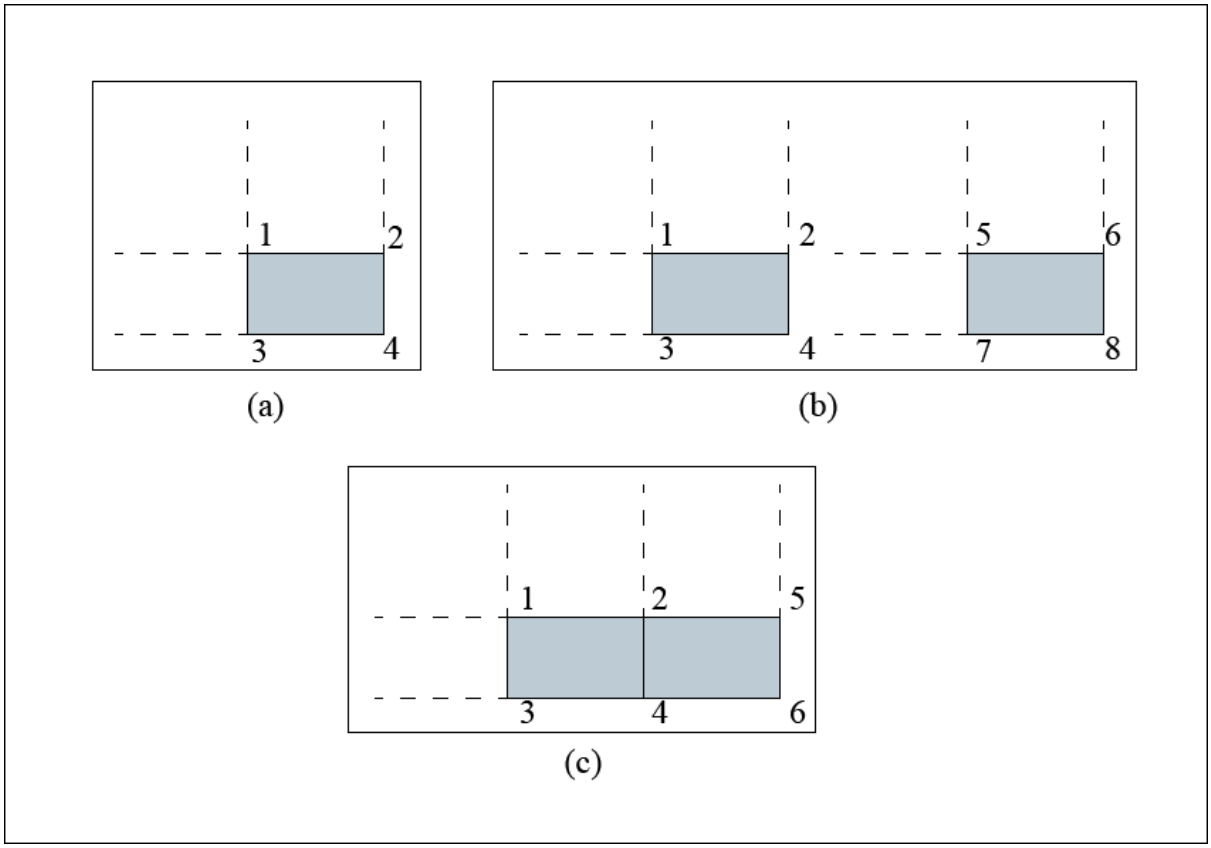

Figure 4: Integral image array reference (a) the sum within the dark shaped location is computed as $4+1-(2+3)$ (four array references) (b) the sum within the two dark shaped location is computed as $4+1-(2+3)$ and $8+5-(6+7)$ respectively (eight array references) (c) the sum within the two adjacent dark shaped location is computed as $4+1-(2+3)+6+2-(4+5)$, hence $6+1-(3+5)$ (six array references).

Cascade

After AdaBoost shortens the number of classifiers, there is still a lot of classifiers to process. To minimize that, cascading is employed. All the features are joined to check a face region as shown in (Figure 5). If first stage of the cascade is matched then it will move to next or reject. A positive outcome from one classifier triggers the next one. Other stages are calculated in the same

procedure.

MATERIALS AND METHODS

Design of surveillance robot

In the design procedure, we used SolidWorks version 2015 as our design tool. At first, a box shaped frame was made containing minimal weight. Secondly, wheels were added in such a way that the robot can move in any desired direction. Finally, a camera holder was designed to give 180 degrees movement flexibility. The designed lightweight and minimalist frame is shown in (Figure 6).

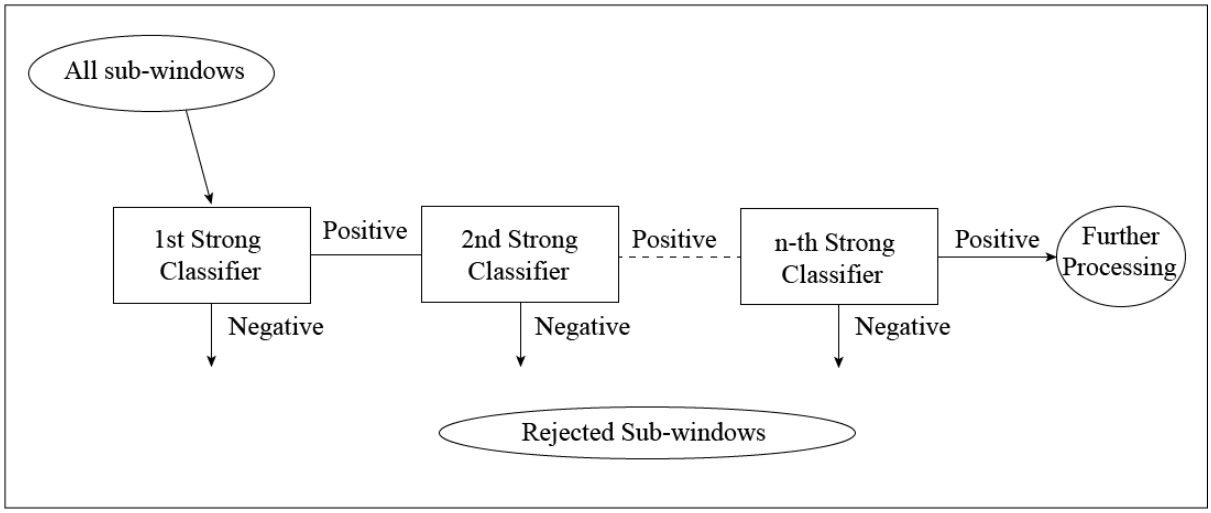

Figure 5: Schematic diagram of the detection cascade. All human faces contain eyes, mouth, etc. These features are taken as a strong classifier. Without strong classifier a frame does not contain a face. That is why they are checked by every strong classifier and rejected at absence. If all the strong classifiers are present in an image, the image is classified as a face.

Fabrication of Surveillance Robot

We used plywood to make the frame so that it has a strong structure to move in any type of environment yet light weight. Furthermore, the camera holder where the camera moves 180 degrees around was made of ply wood which gives it stability and a sustainable movement.

Working Process of Surveillance Robot

We used raspberry pi which has the microprocessor, memory, wireless radios, and ports all on one circuit board. Python programming language was utilized in our system building of the intelligent surveillance robot.

- Step 1: Establish connection with user device via Wi-Fi
- Step 2: Receive user transmitted information
- Step 3: Moves according to the user defined path

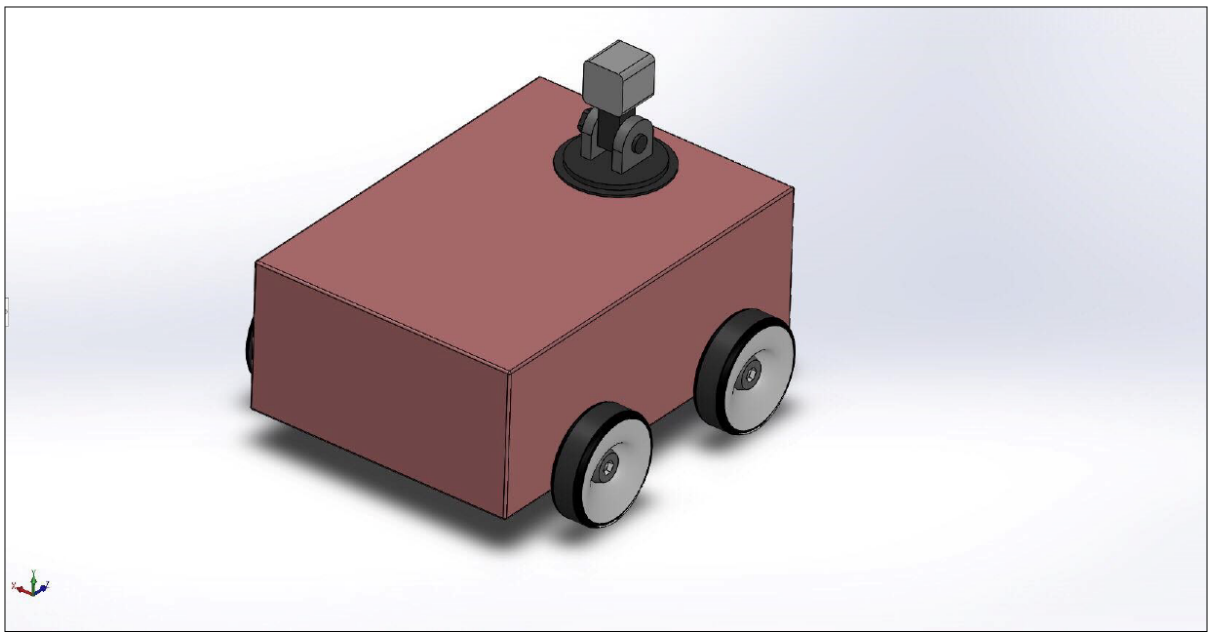

Figure 6: Frame of the surveillance robot. The front side of the frame contains the camera holder. Camera holder allows the camera to turn around 180 degrees. This movement can be controlled by the user from the GUI.

- Step 4: Transmit real time video
- Step 5: Detects Human face if human face comes over its frame

Flow Chart of Face Detection

The process of detecting face also includes showing the results in the GUI. A laptop is connected to the Raspberry Pi with Wi-Fi. The Raspberry Pi receives the video feed captured by the camera. The GUI displays the video stream, which is then analyzed to detect faces and presents on GUI display at the same time. The flow chart of the face detection process is shown in (Figure 7).

RESULTS AND DISCUSSION

Robot Structure

We measured the height, width and length with meter ruler. We also calculated robot's weight with weighting machine. We took the robot out into the open field to test performance. We calculated the time for a certain distance and the time it took to cover that distance with stopwatch. Then we calculated the maximum velocity of the robot and the minimum velocity in cm/sec.

- Maximum velocity of the robot was 83.33 cm/sec and minimum velocity was 23.33cm/sec.
- Measured height = 37cm, width = 26cm , length = 14cm, weight = 1.87 Kg of the surveillance robot.

DOI: https://doi.org/10.53808/KUS.2022.ICSTEM4IR.0167-se

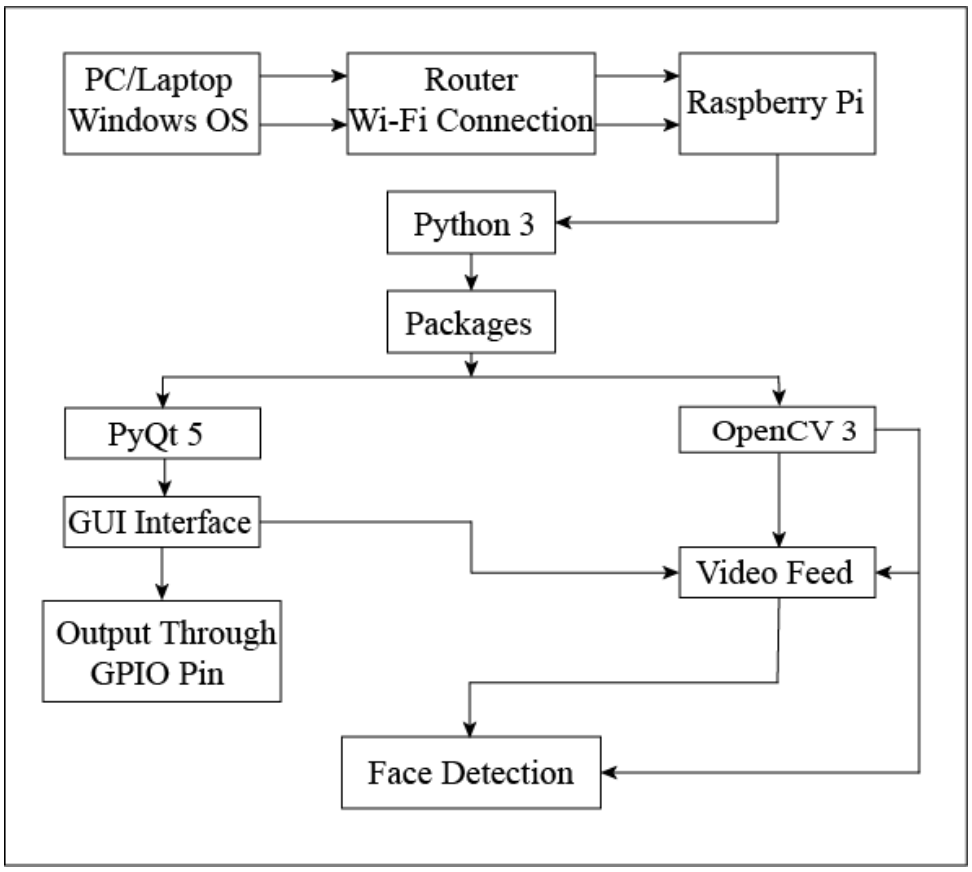

Figure 7: Flow chart of the face detection process. The whole process can be controlled from one or multiple laptops/PCs by a secure Wi-Fi connection. The raspberry is connected with the laptops/PCs via the Wi-Fi and the GUI can be viewed from the user end and the whole process can be monitored and controlled from the user interface.

Graphical User Interface (GUI)

As our aim was to develop a robot which would be user friendly, an easy interface was a must need. We used 'PyQt5' to design our user interface. PyQt5 is a set of Python bindings for Qt5 application framework from Digia. PyQt5 is implemented as a set of Python modules. The designed GUI is shown in (Figure 8).

Face Detection

Our proposed framework successfully detects single faces captured by camera as shown in (Figure 9). The process is size invariant. Additionally, the method tracks human face movement continuously.

While there is multiple faces available in the scene, the method efficiently detects faces as shown in (Figure 10). There is no prior requirement for a declaration of the number of faces that will be present. Slightly tilted faces and variation in scale of the face size also does not contribute in reducing performance.

Over and above that, the framework detects faces with closed eyes as shown in (Figure 11).

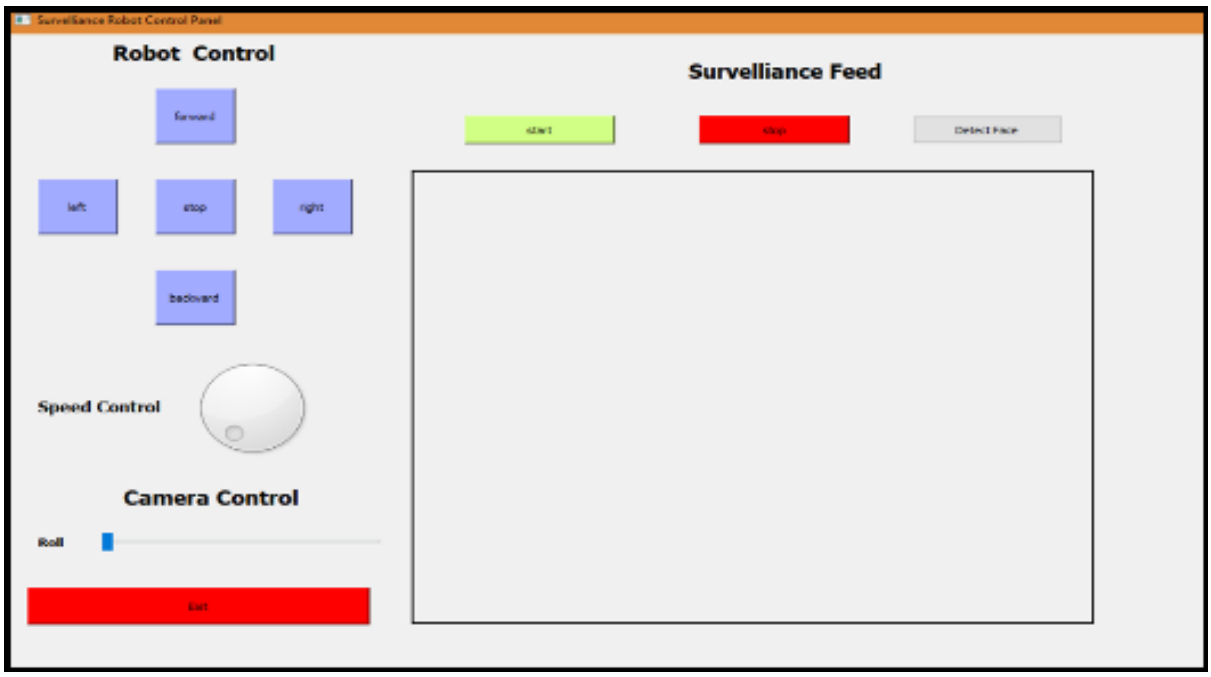

Figure 8: GUI of the intelligent surveillance robot. The GUI is very simplistic and user friendly and contains basic options with easy to recognize and operable interface.

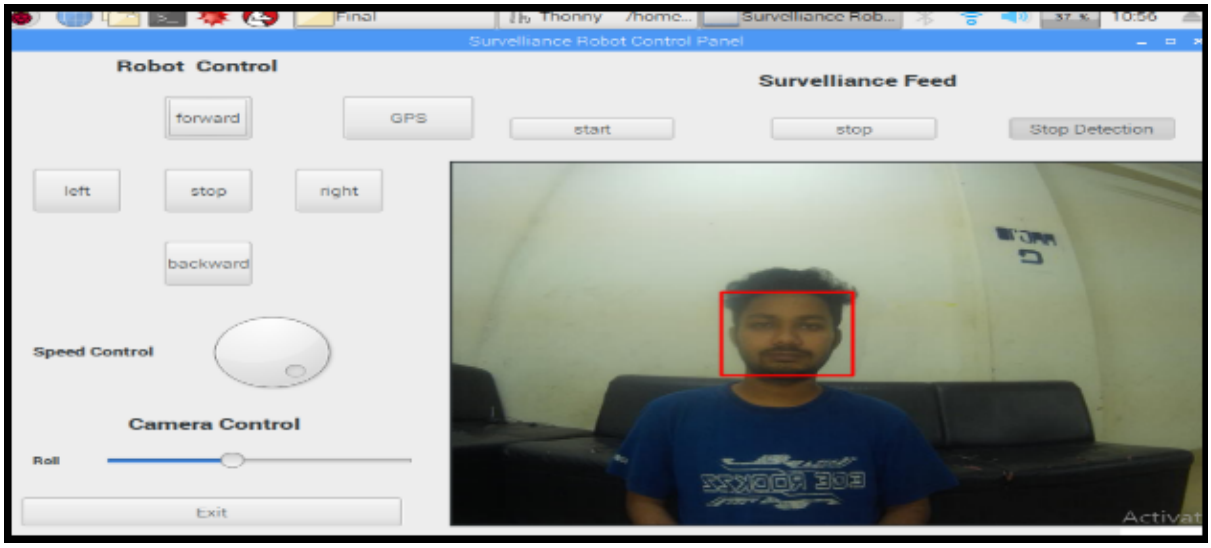

Figure 9: Face detection of single face.

Frontal faces with less than 45 degrees tilt and rotation on either sides is favorably detected.

Experimental Results

Two experiments including the databases Yale Face Database (**P.N.Bellhumer1997**), Natural Images database (Roy et al., 2018), MIT-The Center for Biological & Computational Learning Laboratory (MIT-CBCL) (Weyrauch et al., 2004) and MIT-Indoor Scene Database (Quattoni and Torralba,

DOI: https://doi.org/10.53808/KUS.2022.ICSTEM4IR.0167-se

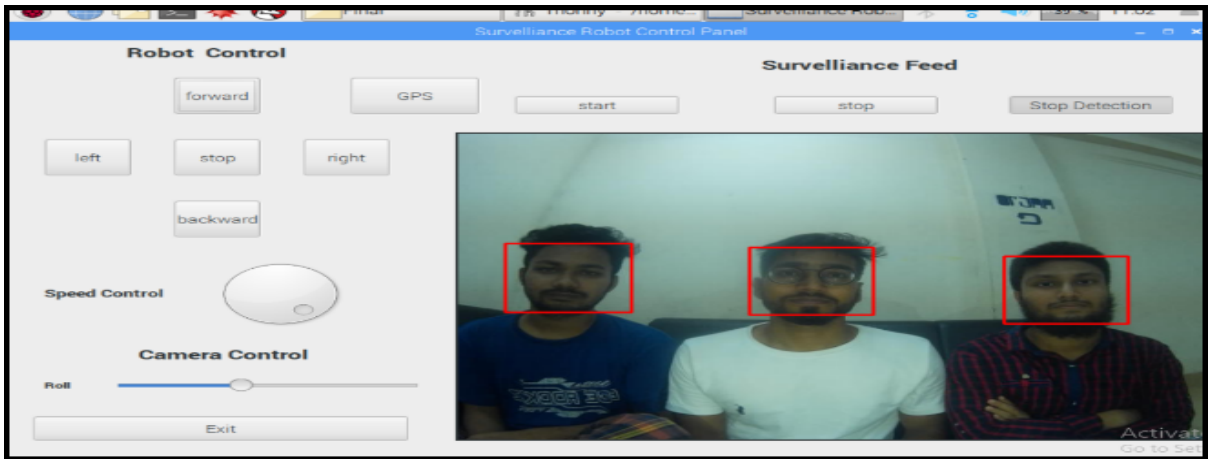

Figure 10: Face detection of multiple faces.

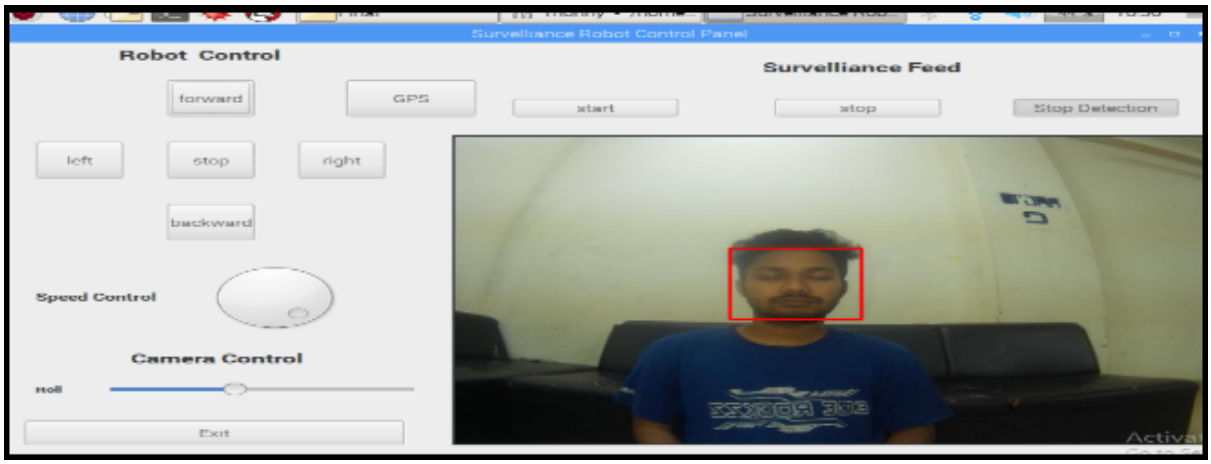

Figure 11: Face detection with closed eyes.

2009) were used to evaluate the system. For the first experiment, the images containing faces and the non-face images were joined from both the Yale and Natural Image databases randomly and on equal number yielding a total sample size of 2373. For the second experiment, the same procedure was followed with MIT-CBCL and MIT-Indoor scene databases yielding a total sample size of 4231. (Table 1) describes the detection accuracy rates.

Table 1: Experimental results of face detection.

Databases	The Detection Accuracy Rate			
	Classes		Accuracy	Reference
	Faces	Non-faces		
Yale + Natural Image	1152	1221	98.38%	(**P.N.Bellhumer1997**), (Roy et al., 2018)
MIT-CBCL + MIT-Indoor Scene	2000	2231	88.83%	(Weyrauch et al., 2004), (Quattoni and Torralba, 2009)

(Table 2) represents a comparative study between existing surveillance robots and our proposed system in terms of weight and computational complexity. While measuring computational complexity, two main prospects are taken into account: time complexity and memory complexity. Time complexity is calculated in terms of One billion Floating-point Operations Per Second (GFLOPS) and average number of Frames Per Second (FPS) while memory complexity is estimated in Gigabytes(GB). Additionally, number of pre-processing and use of filters are also major factors for increasing computational complexity. Use of Viola-Jones algorithm minimizes these complexities also.

Table 2: Time and memory complexity of face detection based surveillance robots.

Reference Studies		Weights		
Karishma et al., 2018		6kgs		
Ross et al., 2021		2 kgs		
Our Paper		**1.87 kgs**		
Reference Studies	Algorithm	Time		Memory (GB)
		GFLOPS	FPS	
Phadtare et al., 2021, Phadtare et al., 2021	SSD	45.8	13.3	0.7
Farrajota et al., 2016, Ren et al., 2015	Faster R-CNN	223.9	5.8	2.1
Phadtare et al., 2021, Redmon and Farhadi, 2017	YOLO	34.9	19.2	2.1
Xue et al., 2021, Dai et al., 2016	R-FCN	186.6	4.7	3.1
Our Paper	**Viola Jones**	**0.6**	**60.0**	**0.1**

CONCLUSION

In this paper, we presented a surveillance robot which was light weight. In particular, we introduced a robot with flexible mobility and a surveillance system with real time face detection. The face detection system was implemented using Viola-Jones algorithm. The whole framework was

built using Raspberry Pi operating system and the intelligent surveillance robot was connected to the system through Wi-Fi. For the system maintenance purposes, we introduced a minimalist user friendly GUI. We provided four functions through the GUI: camera rotation (180 degrees), robot speed control, robot movement, and face detection. The future work is to develop a model to recognize facial expressions and behaviour for better scene interpretability.

References

Abudhagir, U. S., Anuja, K., & Patel, J. (2022). Faster rcnn for face detection on a facenet model. In K. Govindan, H. Kumar, & S. Yadav (Eds.), *Advances in mechanical and materials technology* (pp. 283–293). Springer Singapore.

Budiharto, W. (2015). Intelligent surveillance robot with obstacle avoidance capabilities using neural network. *Computational intelligence and neuroscience*, *2015*, 1–5.

Dai, J., Li, Y., He, K., & Sun, J. (2016). R-fcn: Object detection via region-based fully convolutional networks. *Advances in neural information processing systems*, *29*.

Farrajota, M., Rodrigues, J. M., & du Buf, J. M. (2016). Using multi-stage features in fast r-cnn for pedestrian detection. https://doi.org/10.1145/3019943.3020000

Kanagaraj, R., Amsaveni, M. M., Binsha, S., & Chella Keerthana, S. (2022). Raspberry pi-based spy robot with facial recognition. In A. P. Pandian, R. Palanisamy, M. Narayanan, & T. Senjyu (Eds.), *Proceedings of third international conference on intelligent computing, information and control systems* (pp. 29–40). Springer Singapore.

Karishma, A., Krishnan, K. A., Kiran, A., Dalin, E., & Shivaji, S. (2018). Smart office surveillance robot using face recognition. *International Journal of Mechanical and Production Engineering Research and Development*, *8*(3), 725–734.

Liu, Y., Liu, R., Wang, S., Yan, D., Peng, B., & Zhang, T. (2022). Video face detection based on improved ssd model and target tracking algorithm. *Journal of Web Engineering*, *21*(2), 545–568.

Phadtare, M., Choudhari, V., Pedram, R., & Vartak, S. (2021). Comparison between yolo and ssd mobile net for object detection in a surveillance drone. *Int. J. Sci. Res. Eng. Man*, *5*.

Quattoni, A., & Torralba, A. (2009). Recognizing indoor scenes. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 413–420.

Redmon, J., & Farhadi, A. (2017). Yolo9000: Better, faster, stronger. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7263–7271.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*.

Renuka, B., Sivaranjani, B., Lakshmi, A. M., & Muthukumaran, D. N. (2018). Automatic enemy detecting defense robot by using face detection technique'. *Asian Journal of Applied Science and Technology*, *2*(2), 495–501.

Ross, R., Carver, S., Browne, E., & Thai, B. S. (2021). Wombot: An exploratory robot for monitoring wombat burrows. *SN Applied Sciences*, *3*(6), 1–10.

Roy, P., Ghosh, S., Bhattacharya, S., & Pal, U. (2018). Effects of degradations on deep neural network architectures. *arXiv preprint arXiv:1807.10108*.

Salh, T. A., & Nayef, M. Z. (2013). Intelligent surveillance robot. *2013 International Conference on Electrical Communication, Computer, Power, and Control Engineering (ICECCPCE)*, 113–118.

Hasan, M. K. et al. (2022). Design and Development of a Surveillance Robot. *Khulna University Studies*, Special Issue (ICSTEM4IR): 19-32.

Ullah, R., Hayat, H., Siddiqui, A. A., Siddiqui, U. A., Khan, J., Ullah, F., Hassan, S., Hasan, L., Albattah, W., Islam, M., et al. (2022). A real-time framework for human face detection and recognition in cctv images. *Mathematical Problems in Engineering*, *2022*.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, *1*, I–I.

Weyrauch, B., Heisele, B., Huang, J., & Blanz, V. (2004). Component-based face recognition with 3d morphable models. *2004 Conference on Computer Vision and Pattern Recognition Workshop*, 85–85.

Wibowo, M. E., Ashari, A., Subiantoro, A., & Wahyono, W. (2021). Human face detection and tracking using retinaface network for surveillance systems. *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*, 1–5. https://doi.org/10.1109/IECON48115.2021.9589577

Xue, N., Niu, L., & Li, Z. Pedestrian detection with modified r-fcn. In: In *Uae graduate students research conference 2021 (uaegsrc'2021)*. Khalifa University. Abu Dhabi, UAE, 2021, June.

Yang, X., & Zhang, W. (2022). Heterogeneous face detection based on multi-task cascaded convolutional neural network. *IET Image Processing*, *16*(1), 207–215. https://doi.org/https://doi.org/10.1049/ipr2.12344