Embedded Vision Based Cost Effective Tele-operating Smart Robot

Omkar S. Vaidya, Ravindra Patil, Gayatri M. Phade, Sanjay T. Gandhe

Abstract**:** ***In twenty-first century, robots have taken an important place not only in daily human life but also in different working fields. The computer vision has come to be an extensive area of research for real-time visual applications by connecting with an embedded system. The system is considered to be fine only when it is cost effective and have a good accuracy. Todays in score of fields require smart detection and recognition of objects, collect information from surroundings and perform different tasks such as security system, vehicle surveillance navigation, autonomous robot navigation etc. This paper represents a cost effective robot which is able to recognize different faces, speak their identities using python TTS and track the human smartly with obstacle avoidance feature in real-time. Here OpenCV library for performing different imaging operations and python language at its backend with multi-threading concept is used. Sensors are used to read surrounding parameters and the Wi-Fi (Wireless Fidelity) is the wireless medium to control robot from remote location via android mobile phone by the user. Alexa voice service is used which actually gave liveliness to the robot and is the great experience of intercommunication and joyful task.***

Index Terms**:** ***Embedded Vision, Face Recognition, Human Tracking, Python TTS.***

I. INTRODUCTION

Embedded vision is the absorbing of two technologies as embedded system and computer vision. The systems based on microprocessor generally known as embedded systems and digital image processing with an intelligent algorithm to return meaningful information from images and videos known as computer vision. Embedded vision is now developing across a score of fields consisting of autonomous medical care, agricultural technology, search and rescue situations repair in condition dangerous to humans, operate in unmanned area, ware houses, military surveillance tasks etc. The embedded vision based robot consists of camera, sensors to collect information of its neighboring and microprocessors etc. These components with respective computer vision algorithms detect, recognize objects and track them by calculating their position.

For achieving the real time approach, select the hardware and software with imperative specifications in respective domains. There are a number of "Single Board Computers" available for IOT and embedded platform but Raspberry Pi 3B+ is a powerful and cost efficient SBC and is newer version with Broadcom BCM2837B0, Cortex-A53 64-bit system on-chip at 1.4GHz processor for fast computation and Broadcom Videocore-IV GPU for parallel computing of video data at low clock speed [1]. It is our first client used for computer vision algorithms. For making system more efficient, Raspberry Pi Zero W as a second client for connecting supplementary parts is used and enables Alexa Voice Service. Both clients are connected to the server for us its android mobile. OpenCV library is used for computer vision and python language for coding due to its simple syntax. Sensors such as temperature and humidity sensor, light intensity sensor collect the neighborhood information and send this information to android mobile phone. Robot is controlled by android mobile application through Wi-Fi.

The paper is presented into six sections. The Section II gives the literature survey and Section III explains the proposed system with its hardware and software components. In the Section IV, the workflow design of the system is made clear. The experimental results are presented in Section V and paper is concluded in Section VI.

II. LITERATURE SURVEY

A. Face Recognition

Face recognition is different process than face detection. In face detection, it is the simple process of detecting the face presence in an image or video stream but face recognition is the process of identifying whose the face belongs to. Basically, there are two steps for face recognition as per its definition. In the first step, by using method like Haar cascades, HOG with Linear SVM as a machine learning technique, deep learning one has to detect occurrence of faces in an image or video-stream. The second step is to identify the faces from first localization step and then the name assigning step. The Goldstein, et al. developed first crude approach for face recognition using 21 subjective facial features, such as hair colour and lip thickness, to identify a face in an image. But these 21 measurements were manually calculated and that was the biggest drawback of this approach [2]. OpenCV has three built in face recognizers as 'Eigenfaces', 'Fisherfaces' and 'Local Binary Pattern Histogram'. Kirby and Sirovich presented work on a low-dimensional procedure for the characterization of human faces which is also known as 'Eigenfaces Algorithm' [3].

Revised Manuscript Received on May 10 ,2019

Omkar S. Vaidya, Department of Electronics & Telecommunication Engineering, Sandip Institute of Technology & Research Centre, Nashik, Savitribai Phule Pune University, Maharashtra, India.

Ravindra Patil, Department of Electronics & Telecommunication Engineering, Sandip Institute of Technology & Research Centre, Nashik, Savitribai Phule Pune University, Maharashtra, India.

Dr. Gayatri M. Phade, Department of Electronics & Telecommunication Engineering, Sandip Institute of Technology & Research Centre, Nashik, Savitribai Phule Pune University, Maharashtra, India.

Dr. Sanjay T. Gandhe, Department of Electronics & Telecommunication Engineering, Sandip Institute of Technology & Research Centre, Nashik, Savitribai Phule Pune University, Maharashtra, India.

This standard linear algebra technique for dimensionality reduction called as Principle Component Analysis (PCA) which constructs a low-dimensional depiction of face images. It identifies face using a feature vector smaller than 100-dim. Kirby and Sirovich represented the linear algebra technique such as Linear Discriminant Analysis for face recognition and that was the next research. The Timo Ahonen, et al. proposed technique to divide face image in to a 7 x 7 grid of equally sized cells. After this faces were extracted the LBPH from 49 cells and the concatenation of weighted LBPH formed the final feature vector. The k-NN classification is used for actual face recognition [4].Manop Phankokkruad, et al. studied the face recognition accuracy affected by facial expressions and viewpoint variations for OpenCV face recognizers. The experimental results show that LBPH is the most precise algorithm for real world implementation which achieved 81.67% of accuracy in image-based testing. For Eigenfaces the accuracy is 46.67% and for Fisherfaces, it is 69.33%. Also with variation in facial expressions, LBPH showed accuracy of 79.69% in the normal case, 80.30% in smile case, and 80.95% in grin case [5]. Umm-e-Laila, et al. compared LBPH, Eigenfaces and Fisherfaces algorithms for speed and accuracy. Finally, research work conclusion was given as LBPH has 90% accuracy rate in comparison with other two and Raspberry pi can be an important component for development of different biometric security systems [6].

B. Human Detection

Sanyam Mehta, et al. caught the attention to the point that in the present days, there are different kinds of objects available with dissimilar identities and having different attributes such as a colour, size, texture, volume etc. The object detection is the procedure of finding presence of particular object in video streams and images [7]. In our case the human being as object is going to detect.

Navneet Dalal and Bill Triggs have presented an approach for human detection. HOG features are an excellent for describing object shape and good for human detection whereas HAAR features are good for describing object shading and give first rate result for frontal face detection [8]. Li. Fei, et al. used HOG with linear SVM algorithm for pedestrian detection in intelligent transportation. They tested algorithm on disclosed INRIA database and actual traffic video which achieved the great accuracy for pedestrian detection [9].

C. Object Tracking with Obstacle Avoidance

Object tracking is locating an object in successive frames of a video streams. In OpenCV, there are eight built-in object tracking algorithms such as BOOSTING, MIL, TLD, KCF, MEDIANFLOW, GOTURN, MOSSE and CSRT. All these trackers have different performance and execution accuracy. From these, GOTURN is only one Convolutional Neural Network (CNN) based algorithm in OpenCV. And very important point to note here is, OpenCV 3.4+ versions only consist of all these eight trackers [10]. Widodo Budiharto proposed a tracking mobile robot for surveillance by using Raspberry pi and AVR with obstacles avoidance feature. For this, obstacle avoidance had been achieved with use of ultrasonic sensor which estimate the distance from the object [11]. Mohhamad Rubaiyat Tanvir Hossain, et. al. has suggested an approach for web controlled and partially autonomous vehicle system which minimizes risk of human life and offered the highest safety during driving. The proposed car has the ability to detect traffic light, lane and consists of obstacle avoidance feature by using ultrasonic sensor which calculate the distance and control the motor rotation accordingly [12].

D. Python Text to Speech

In python, there a score of API's available to convert text to speech such as gTTS (Google Text to Speech API), eSpeakNG, pyttsx3 et. The gTTS convert the input text in to audio and saved as mp3 file for speaking purpose. It works only when the internet is connected and it has a very natural sound. The cross platform and offline TTS library is pyttsx3 which works for both python 2 and 3 versions. It supports different speech engines like espeak, sapi5, nsss. The eSpeakNG is an open source, small size and multiple language speech synthesizer supporting python 2 and 3 [13].

III. PROPOSED ARCHITECTURE

In this paper, a smart and cost effective robot is introduced which makes use of Raspberry Pi 3 model B+ as a first client, Raspberry Pi Zero W as a second client and Wi-Fi for secure communication. The block diagram of proposed work is shown in Fig. 1. The principal component is Raspberry pi 3 model B+ to which camera is connected and all computer vision algorithms will be performed by that module. The ultrasonic sensor, light intensity sensor, temperature and humidity sensor are linked to the Raspberry Pi Zero W. These sensors gather imperative information of its neighborhood for considerable accomplishment. The microphone and speaker are connected to the second client for input and output voice for Alexa Voice Service and for python TTS output voice. The motor driving module is also connected to same with its DC geared motor and wheels. Another important aspect is android mobile which provide the manual controlling feature via Wi-Fi from remote location and display the collected information from sensors and camera.

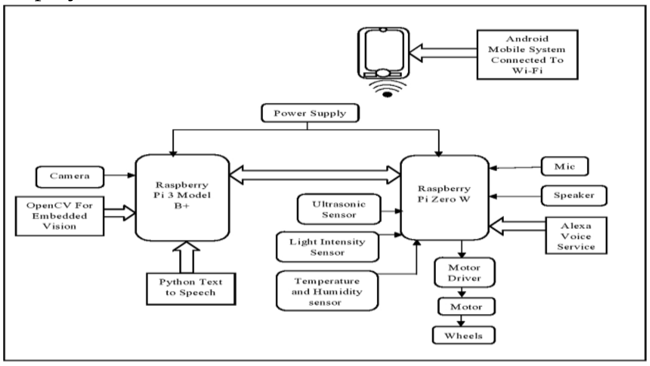

Fig. 1 Block Diagram of Proposed System

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

A. Hardware Components

The below given are the hardware components for the execution of robot:

1) Raspberry Pi: The Raspberry Pi 3 model B+ as client is utilized which has Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC at 1.4GHz processor with 1GB LPDDR2 SDRAM and 40-pin GPIO header. For wireless connectivity it consists of Bluetooth 4.2, BLE as well as 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN. The Broadcom Videocore-IV GPU supports to the parallel computing of video data at low clock speed. Raspberry pi runs the Raspbian stretch operating system and work for running image processing algorithms on OpenCV platform.

2) Raspberry Pi Zero W: In our system, it is client second having single core 1GH CPU, 512MB of RAM, 802.11.b/g/n wireless LAN, Bluetooth Low Energy (BLE) and HAT-compatible 40-pin header. It is used to control the GPIO for sensors, motors and Alexa voice service enabled in it is accessed by microphone and speaker. Its working depends on the commands received from the android application by the user via Wi-Fi.

3) USB Webcam: The Logitech USB webcam is utilized to capture the video for live streaming and processing.

4) Ultrasonic Sensor: The HC-SR04 ultrasonic sensor which helps for obstacle avoidance of the robot path. This sensor gives the confidence to robot for autonomous navigation and continuously suggests the operator to be safe in manual mode of processing.

5) Temperature and Humidity Sensor: DHT22 Temperature and Humidity Sensor are used to measure the temperature and humidity of surrounding in a real time condition.

6) Light Intensity Sensor: To detect the light presence or to measure the light intensity, LDR sensor is used.

7) Microphone: Adafruit mini USB microphone is used for input voice to the Alexa.

8) Speaker: The speaker of 0.5W (8 ohm) for output voice is used for Alexa and python TTS.

9) Motor Driver and Motors: Motozero is an excellent way for controlling motors which is capable of controlling four motors independently. The two Micro-Metal DC Gear motors running at 100 RPM are used for moving of the robot and consume 6V for it.

10) Android Mobile: To control robot from remote location, the operator uses android mobile system. This human controlling technique is usually called as a “Teleoperating” and controlling of the semi-autonomous robots is area of a robotics and known as “Telerobotics”.

B. Software Components

The following stated are the software elements used for running the robot:

1) OpenCV: OpenCV is an open source computer vision library used for a real-time image processing and has a strong computational efficiency.

2) Python: The programming related to the image processing, GPIO control, communication with android application all has been done with python language.

3) Android Application: The android application is developed for controlling the robot from remote location and display the information collected by the sensors.

4) Live video streaming to Android Application: There is wide open source support available to streaming video from Raspberry Pi webcam to android system. Connecting to the IP address in same local network, android system can access the live video stream from Raspberry Pi webcam.

5) Datasets: The face recognition dataset consists of captured face images and trained LBPH face recognizer file for further face recognition task. Another dataset consists of descriptors defining human body features.

6) Alexa: For the intercommunication or just a joyful task is done by using Alexa Voice Service. The python virtual environment is used to give separate place for it and keep system clean and tidy [14].

7) Python TTS: Here eSpeakNG speak synthesizer is used to speak identity of person by name or unknown person.

8) Socket Programming: It is required for connecting clients to the server.

IV. WORK FLOW DESIGN

The flowchart of proposed work is shown in Fig. 2 which clarifies how the hardware and software components of the robot process jointly. The process gets started by powering the system and Wi-Fi is the remote connection for controlling the robot. The webcam of first client detects which type of object is there. If the face comes in front of webcam, then it will recognize the face from stored dataset and speak the “name” using python TTS eSpeakNG and whether the face is unidentified then speak “unknown”.

Another object is human body which is firstly detected and then tracked by the KCF with PID controller from user’s command. The user is capable of controlling motors connected to second client from remote location using android application with live streaming video and ultrasonic sensor alerts continuously from path avoiding obstacles.

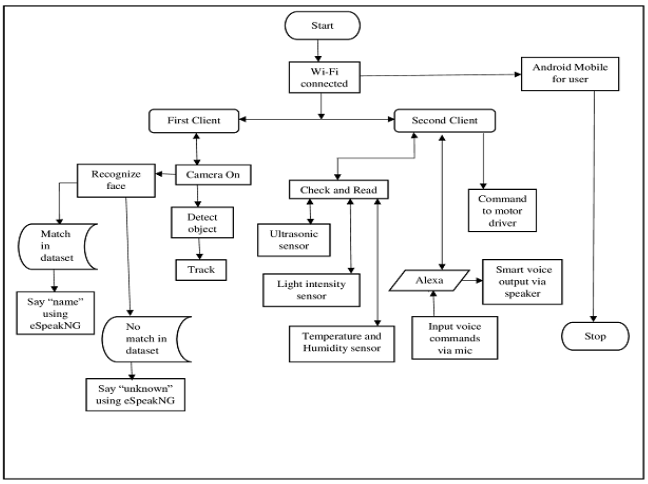

Fig. 2 Flowchart of Proposed System

The Temperature and Humidity sensor, Light intensity sensor collects the surrounding information and sends to the android mobile system to aware us. The very interesting part of robot is Alexa Voice Service which gives liveliness to it.

V. EXPERIMENTATION

A. Multi-threading for Real-time Video Stream

To get real-time result on CPU is very weighty job for video processing applications. While considering about Raspberry Pi, it is a CPU structured SBC and hence the multi-threading concept is used.

Our major thread does both task of reading and decoding of the frames in OpenCV python coding. This is the actual reason for slowing down the whole process. The use of "cv2.VideoCapture()" function and ". read()" method obstruct our major thread to continue processing. To overcome this, another thread is created to read the new frame from web camera while major thread processes only on the current frame. This gives an incredible result for improving frames per second (FPS) and is determined as the total number of read frames in the process divided by total number of seconds of processed time.

The Table I as well as Fig. 3 and Fig. 4 shows the improvement of FPS on processed time in seconds by using multi-threading concept for two different systems such as Raspberry Pi 3 Model B+ and Intel ULV Processor. This improvement in FPS is in effect of reduction in downtime.

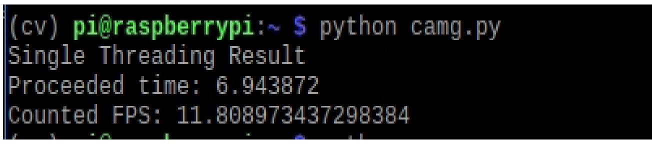

Fig. 3 Single Threading FPS Result

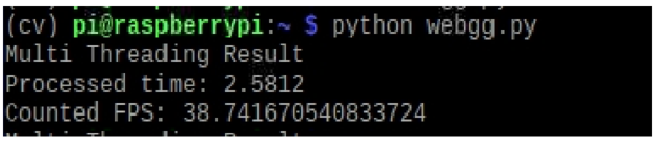

Fig. 4 Multi-Threading FPS Result

Table I. Comparison between Raspberry Pi and Intel Ultra Low Voltage (ULV) Processor

System Type	Raspberry Pi 3 Model B+	Intel ULV Processor
Specification	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz CPU and Broadcom Videocore-IV GPU	Intel(R) Core(TM) i5-3337U CPU @ 1.80GHz and Intel(R) HD Graphics 4000
Operating System	Raspbian Stretch	Windows 10
Processed time in seconds (Single threading)	6.9438	3.1697
FPS (Single threading)	11.8089	18.0898
Processed time in seconds (multi-threading)	2.5812	3.4447
FPS (multi-threading)	38.7416	133.3870
Resolution	400x300	640x480

Finally, at the time of implementation, output displaying cv2.imshow function is removed and cv2.waitKey() which increased our FPS from 38.7416 to 47.3031 at processed time of 2.2290 seconds for Raspberry Pi and 133.3870 to 189.9431 at processed time of 3.3994 second for Intel ULV Processor. The FPS comparison is shown in Fig. 5. To achieve real time results for all further programming, multi-threading concept is used.

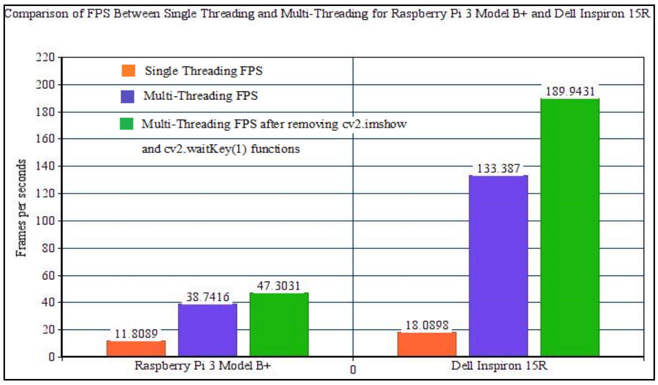

Fig. 5 Graphical Comparison of FPS between Single Threading and Multi-Threading for Raspberry Pi 3 Model B+ and Intel ULV Processor Using Webcam

B. Face Recognition using LBPH Algorithm

As discussed in section II, the face recognition is a two-step process which includes face detection and then recognition means identity that person. In our case, Haar cascade classifier is used for face detection which works on natural structure present in the faces. These extracted features are used to learn generic models and detect the faces. The classifier is learned by providing faces and non faces images. By using convolutional kernels HAAR-features are extracted from images and calculate feature values. Whether the feature value is above user computed threshold then it is considered as match otherwise not. The result of real-time face detection using Haar cascade on OpenCV platform is shown in the Fig. 6.

Fig. 6 Face Detection using Haar Cascade

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

The implemented system uses LBPH algorithm in OpenCV for face recognition. The dataset of tagged faces has been created for training the recognizer as shown in the Fig. 7. The system is tested with different number of tagged face images in the dataset and preferred 18 faces for each person who gave best results.

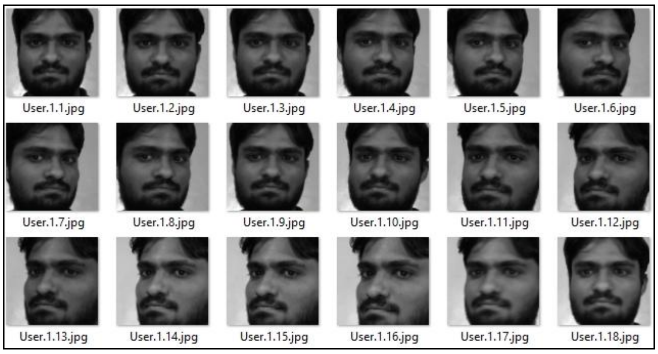

Fig. 7 Dataset of Tagged Faces

The face image is divided in to the local same sized cells and Local Binary Pattern Histogram is extracted from each cell. The centred cells carry more facial information than the corner cells. All the maximum weighted LBPH are gathered to build feature vector. Finally, the k-NN classification does the face recognition by utilizing x2 distance between the input face image and tagged faces in the dataset. The real-time results with following two frames for face recognition are as shown in the Fig. 8 and Fig. 9.

Fig. 8 Face Recognition for tagged face in dataset and another unknown face using LBPH Algorithm

Fig. 9 Face Recognition for both tagged faces in dataset using LBPH Algorithm

The Detection time and FPS during face recognition process are counted on both models. From Table II and graph shown in Fig. 10, it is clear that newer model of Raspberry Pi gives better result than previous version.

Table II. Comparison of Detection Time and FPS count

System Type	Raspberry Pi 3 Model B+	Raspberry Pi 3 Model B
Specification	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz CPU and Broadcom Videocore-IV GPU	Broadcom BCM2837, Cortex-A53 (ARMv8) 64-bit SoC @ 1.2GHz CPU and Broadcom Videocore-IV GPU
Detection time in microseconds	670	790
Frames per seconds	9.890	7.943
Resolution	400x300	400x300

C. Human Detection and Tracking

The human detection is achieved by using HOG descriptor and SVM (Support Vector Machine) detector. HOG descriptor is extracted by sliding window technique and then SVM classifier detects whether the captured frame contains human or not. If the detection probability is higher than bounding box is recorded. The non-maximum suppression is applied to remove excessive and overlapping bounding boxes which show the following result as shown in Fig. 11.

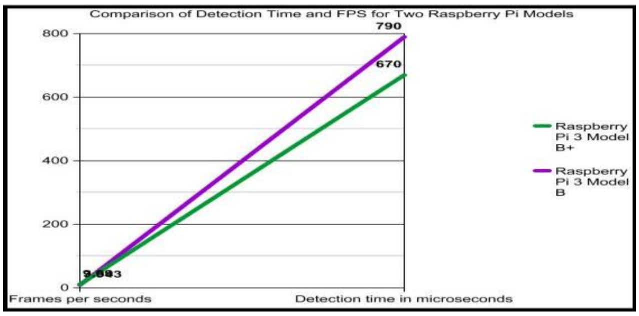

Fig. 10 Graphical Comparison for Detection Time and FPS between Two Different Raspberry Pi Models during face recognition

Fig. 11 Result of Complete Human Body Detection

For the real-time human detection hog.detectMultiscale parameters such as frame, winStride, padding and scale are very important for maintaining the trade-off between speed and accuracy which is computed and shown in Table III and IV.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

The very first, frame size should be as small as possible for minimum data processing with maintaining detection accuracy. The padding tupple parameter contributes for increasing the accuracy of a detector for our case, accuracy is obtained at (16, 16). Further two parameters as winStrand and Scale are set to (8, 8) and 1.19 respectively.

The number of levels in image pyramid are affected by scale parameter where as "step size" is oppressed by winStrand parameter in both x and y position of the sliding window. These parameters reduced the false positive detection rate and increased our FPS to 4.76 as compared to NVIDIA Jetson TK 1 CPU board's FPS of 3.5 as shown in Table V [15]. After removing cv2.imshow and waitKey(1) functions, 5.44 FPS is obtained which is represented in Fig. 12.

Table III Real-Time Results of varying Winstride Parameter

WinStride	Detection time in seconds	Detection Accuracy	Frames per seconds
(4, 4)	0.4261	Good	1.21
(8, 8)	**0.1906**	**Good**	**3.30**
(16, 16)	0.1346	Not Good	3.70

Table IV Real-Time Results of varying Scale Parameter

Scale	Detection time in seconds	Detection Accuracy	Frames per seconds
1.01	1.187	Good	0.70
1.03	0.4478	Good	1.73
1.05	0.2204	Good	2.69
1.07	0.1872	Good	2.76
1.09	0.1s642	Good	3.62
1.11	0.1482	Good	3.94
1.17	0.1384	Good	4.72
1.19	**0.1342**	**Good**	**4.76**
1.21	0.1149	Not Good	4.97
1.3	0.1183	Not Good	5.30
1.4	0.1143	Not Good	5.38
1.5	0.1140	Not Good	5.47

Table V Varying Winstride Parameter Result

System Type	Raspberry Pi 3 Model B+	NVIDIA Jetson TK1
Specification	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz CPU with Broadcom Videocore-IV GPU, 1GBLPDDR2 RAM	4*2.3GHz ARM15 CPU, TEGRA K1 GPU, 2GB DDR3 RAM
Operating System	Raspbian Stretch	Ubuntu
Frames per seconds	4.76	3.5
Price	35$	99$

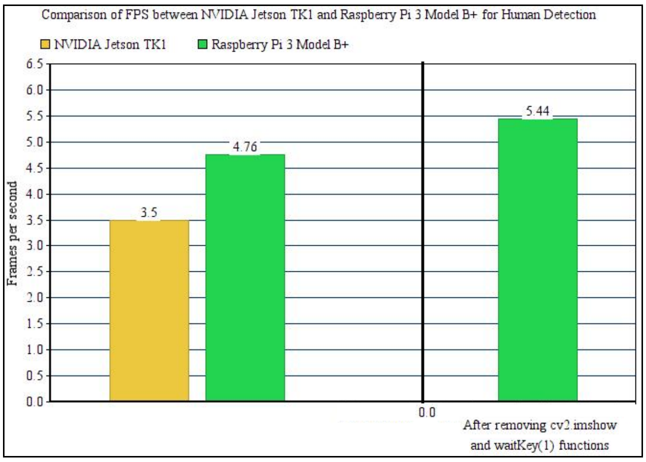

Fig. 12 Graphical Comparison of FPS between NVIDIA Jetson TK1 and Raspberry Pi 3 Model B+ for Human Detection

For tracking, the OpenCV built-in KCF tracker with PID controller is used which continue velocity and heading of the robot.

D. Android Mobile Application

As shown in Fig. 13, there are two screens for mobile application from which the first screen is for user's command and second for displaying temperature, humidity and light intensity parameters got from the sensors. After connecting to the respective IP address, the live video stream is received from webcam to our mobile. The basic commands will be given via Wi-Fi connection such as go (front), back, left, right and autonomous mode is started by start command and stopped by stop command. Also speed is adjusted by the user as per requirement.

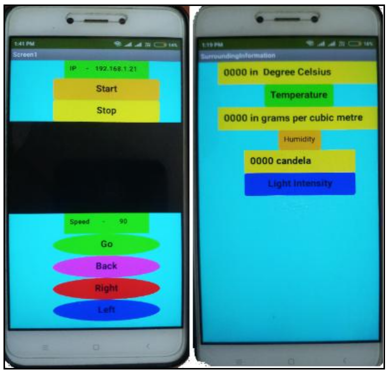

Fig. 13 Android Mobile Application for user to control robot

E. Alexa and eSpeakNG

Alexa voice service has been enabled in Raspberry Pi Zero W for an intercommunication purpose. The input voice commands are given by the mic and get the answered voice from speaker. The eSpeakNG give the voice output via speaker by speaking "name" or "unknown" identity of that person.

VI. CONCLUSION

In this robotic edge of world, a proposed smart robotic system is capable to develop embedded vision tasks according to essentials in a pocket worth. The proposed cost effective robot performs the tasks such as identify the person, speak the name, track the human and communicate with a user by using embedded vision. The accuracy and speed has been achieved as per expectations by exploring the basics and effective coding. By using multi-threading, the speed (i.e. FPS of real-time videostream) is improved by 35.4942 FPS (i.e. 300.57% increased) for Raspberry Pi. Also for real-time human detection, 1.94 increased FPS is obtained (i.e. 55.43% of improvement over the results of NVIDIA Jetson TK1 CPU with good accuracy). Overall the system can be deployed for home or society, security purpose, medical care, warehouses, reception counters, schools and colleges for attendance etc.

REFERENCES

1. [Online] Raspberry Pi 3 Model B+ Product-sheet Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/
2. A. J. Goldstein, L. D. Harmon and A. B. Lesk, “Identification of Human Faces”, Proceedings of IEEE, Vol. 59, Issue 5, pp. 748-760, May 1971, DOI: 10.1109/PROC.1971.8254.
3. L. Sirovich and M. Kirby, “Low-dimensional procedure for the characterization of human faces”, Journal of the Optical Society of America A, Vol. 4, pp. 519-524, March 1987, DOI: 10.1364/JOSAA.4.000519.
4. Timo Ahonen, Abdenour Hadid, and Matti Pietik¨ainen, “Face Recognition with Local Binary Patterns”, European Conference on Computer Vision, pp. 469-481, Prague, Czech Republic 2004. DOI: 10.1007/978-3-540-24670-1_36.
5. Manop Phankokkruad and Phichaya Jaturawat, “Influence of facial expression and viewpoint variations on face recognition accuracy by different face recognition algorithms”, IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Kanazawa, Japan, 2017. DOI: 10.1109/SNPD.2017.8022727
6. Umm-e-Laila, Muzammil Ahmad Khan, Muhammad Kashif Shaikh, Syed Annas Bin Mazhar and Khalid Mehboob, “Comparative analysis for a real time face recognition system using raspberry Pi”, IEEE International Conference on Smart Instrumentation, Measurement and Application (ICSIMA), Putrajaya, Malaysia, 2017. DOI: 10.1109/ICSIMA.2017.8311984
7. Sanyam Mehta, Arindam Misra, Ayush Singhal and Praveen Kumar, “Parallel implementation of video surveillance algorithms on GPU architectures using CUDA”, IEEE International Conference on Advanced Computing and Communications (ADCOM), 2009.
8. N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection”, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, 2005, DOI: 10.1109/CVPR.2005.177.
9. Li Fei and Li Deng Lin, “Pedestrian Detection Based on Histogram of Oriented Gradient in Intelligent Transportation System”, International Conference on Intelligent Networks and Intelligent Systems (ICINIS), Shenyang, China, 2013. DOI: 10.1109/ICINIS.2013.27
10. [Online] Open Source Computer Vision Tracking API Classes Available: https://docs.opencv.org/4.0.0/d9/df8/group__tracking.html
11. Widodo Budiharto, “Design of tracked robot with remote control for surveillance”, International Conference on Advanced Mechatronic Systems, Kumamoto, Japan, 2014. DOI: 10.1109/ICAMechS.2014.6911568
12. Mohammad Rubaiyat Tanvir Hossai, Md. Asif Shahjalal and Nowroz Farhan Nuri, “Design of an IoT based autonomous vehicle with the aid of computer vision”, International Conference on Electrical, Computer and Communication Engineering (ECCE), 2017, Cox's Bazar, Bangladesh, DOI: 10.1109/ECACE.2017.7913003.
13. [Online] Python interface for eSpeak NG Available: https://pypi.org/project/py-espeak-ng/
14. [Online] Amazon Alexa Voice Service (AVS) Tutorials Available: https://developer.amazon.com/docs/alexa-voice-service/set-up-raspberry-pi.html
15. Sumitra Pundlik, Vikrant Patil, Ashwin Sahasrabudhe, Shweta Shinde, “Object tracking bot using on-board Jetson TK1: An approach to reduce communication overhead and time delay”, Department of Computer Engineering, MIT College of Engineering, Pune, India, DOI: 10.1109/BID.2017.8336575.

AUTHORS PROFILE

Omkar S. Vaidya received B.E. degree in 2009, M.E. degree in 2013 and currently pursuing Ph.D. He is an Assistant Professor in E&TC Dept. of Sandip Institute of Technology & Research Centre, Nashik. He is a life member of ISTE, IETE, IEI, ISCA, IAENG and IRED. He has published 18 papers in International Journals and presented 24 papers in National/International Conferences. He received Best Paper Award twice in conferences. He also received Research Grant from Savitribai Phule Pune University, Maharashtra. His research interests are Image Processing and Computer Vision.

Ravindra Patil is student of Master of Engineering (M.E.) at Sandip Institute of Technology & Research Centre, Nashik. Prior to this he completed the graduation B.E. (E&TC) in 2017. His recent work is focused on Computer Vision and Machine Learning.

Dr. Gayatri M. Phade is an Associate Professor and Head of E&TC Dept. at Sandip Institute of Technology & Research Centre, Nashik. She received B.E. degree in 1998, M.E. degree in 2006 and Ph.D. in 2016. She is life member of IEEE, ISTE, IETE and fellow member of IEI. She has published 23 research papers in International Journals and presented 22 papers in National/International Conferences. She has received Research Funding from National Entrepreneurship Board. She is recipient of Women’s Achiever Award and Best Teacher Award. Her area of expertise is FET Design and Modeling, Internet of Thing and RFIC Design.

Dr. Sanjay T. Gandhe is Principal of Sandip Institute of Technology & Research Centre, Nashik. He completed B.E., M.E. and Ph.D. (VNIT, Nagpur) in 1990, 1996 and 2008 respectively. He has published 27 research papers in International Journals and presented 41 papers in National/International Conferences. He is fellow member of IETE, IEI, Senior Member of IEEE and life member of ISTE. He has fetched several Research Grants from Governmental Apex Bodies. He is honored as Best Principal Award and Best Teachers in Professional Educational Category. He was Reviewer of IEEE Transaction on Dependable and Secure Computing in 2011-12. He served as Session Chair in various National and International Conferences. His area of expertise is Image Processing, Robotics and Machine Vision.