Journal of Information Science and Engineering 36, 1055-1067 (2020)
DOI: 10.6688/JISE.202009_36(5).0008

Interactive Robotic Testbed for Performance Assessment of Machine Learning based Computer Vision Techniques*

Nithin P. B., Albert Francis R., Ajai John Chemmanam,
Bijoy A. Jose and Jimson Mathew+
Cyber Physical Systems Lab, Department of Electronics
Cochin University of Science and Technology
Kerala, 682022 India
E-mail: {nithinpb180; albertfrancis32632}@gmail.com; {ajaichemmanam; bijoyjose}@cusat.ac.in
+Department of Computer Science and Engineering
Indian Institute of Technology Patna
Bihar, 801103 India
E-mail: jimson@iitp.ac.in

Computer vision, a widely researched topic over the years, got a shot in the arm with the arrival of high performance and cloud computing. Online and offline techniques for object detection, recognition and tracking have a huge impact in real-world applications such as video surveillance, biometric authentication and targeted advertising. With machine learning, conventional feature extraction based implementation has given way to the model based implementations. This demands high compute speed to keep up with complex trained models. Computer vision with Machine learning solved some of the traditional problems like image classification and is now offering new unique problems in image processing such as object tracking, object segmentation *etc*. Performance assessment of various computer vision applications in object tracking, when used with machine learning solutions, is a high priority. With this intent, we propose a robotic testbed for various computer vision applications such as face recognition, tracking, gesture detection, character recognition, *etc*. It has a hardware tracking system based on face detection and recognition. A fully functional robot with a table lamp design is made to work with these applications using multiple algorithms and their performance parameters are compared. Since a low compute power setup is used, the robot can work properly only on optimized implementations. Visual intelligence to recognize gestures and capability to read text were integrated onto the robot.

Keywords: face detection and tracking, performance analysis, computer vision, machine learning, neural network, gesture recognition, optical character recognition

1. INTRODUCTION

Recent advancements in machine learning and artificial intelligence have made significant progress in computer vision and its applications. The ability for analyzing a large amount of data and pattern recognition has made the area popular among data analysts as well as researchers. Robotics, which studies the possibility of human-machine interaction (HMI) is one of the major fields that got benefited with these advancements. Humanoid

Received August 10, 2019; accepted September 26, 2019.
Communicated by Ashutosh Kumar Singh.
* This work is supported by DST-SERB-ICPS project titled "Energy Efficient Cyber Security Implementation for Internet of Things", with Dr. Bijoy as principal investigator.

robots that act as personal assistants are not a character of science fictions anymore. As these robots are given tasks to assist people in their works, the robots need to interact and communicate with people around them. The ability to detect and identify the faces is a primary requirement for robots to enable personalization. Face analytics is challenging because the facial features tend to change with age, gender, ethnicity, illumination, facial expression, orientation, makeup and use of accessories on the face.

A lot of algorithms have been developed with good accuracy for the facial analysis over the last two decades. The HAAR cascade detector, Histogram of Oriented Gradients (HOG) detector and Local Binary Pattern (LBP) detectors are the major techniques used until the evolution of neural networks. Most of the algorithms work fairly on frontal faces but fails to perform when the faces are at extreme angles. Arunmozhi *et al.* [1] compared the performance of different feature extractors like HAAR, HOG [2] and LBP [3]. HAAR works accurately well for frontal faces in illuminated facial images, The performance takes a hit when dealing with low light/ under natural lights. The LBP features extractor is good at capturing local patterns. On the other hand, HOG feature extractor is great at capturing edges and corners in images and the number of false positives is comparatively less. In [4], authors have developed a robot that can interact with people around it. They used the Viola-Jones algorithm with Haar-like features for face detection and uses the KLT algorithm [5] for face tracking. A robot with 2 degrees of freedom was used for face tracking purpose. Dang [6] also compares the performance of various face detection algorithms such as Viola-Jones Face Detection Algorithm, Successive Mean Quantization Transform(SMQT) [7] Features and Sparse Network of Winnows(SNOW) [8] Classifier Method, Neural Network based Face Detection and Support Vector Machines. The authors concluded that Viola-Jones face by Paul Viola *et al.* [9] has the best performance among them, followed by SMQT Features and SNOW Classifier Method than Neural Network for facial detection. With recent breakthroughs, these findings might have become obsolete. Technology majors such as Google, Microsoft, Amazon, IBM, *etc.* are providing face analytics as a service through their cloud platforms with a neural network backend. Considering this, We decided to assess and compare the performance of traditional methods with Caffe Deep Neural Network models [10].

From biometric verification to identifying thieves using video surveillance, several face detection and recognition applications exist. Face verification for login onto portable devices is common nowadays and is much more secure than remembering and typing a pin or password. In photography, face detection is the base for smile detection and selfie portrait mode technologies. It also helps for automatic image database management, by grouping images of the same persons. Even advertisers are now moving onto targeted advertisement, method of showing ads specifically to users based on their age, gender, ethnicity *etc*. In video surveillance, any occlusion to tracking object, either intended or unintended, can cause the tracking to be lost when using static cameras. Through this work, we try to build and test a robotic arm that follows the faces detected. Enhanced tracking with hardware and software support, helps the system to adjust with its environment. Further face recognition helps the tracking bot to keep the focus on the recognized face and do surveillance.

As an active tracking robot with a camera, adding visual intelligence improves its capabilities. Primary visual intelligence includes detecting objects in the images and reading texts from them. In this work, we try to detect hands and can be used for gesture detection to make the bot more user interactive. [11] compared the performance of various algorithms such as Adaptive Boosting Algorithm [12], Graham's Scan Algorithm *etc*. They claimed to have achieved 92% accuracy with Graham's Scan Algorithm and 70% accuracy with Adaboost Learning. Reading texts includes two parts, understanding the text from images with the help of Optical Character Recognition (OCR) engine, and synthesis audio for the text understood. In [13], A comparison of the performance of various open-source OCR engines such as Calamari, Tesseract 4, OCRopy, OCropus 3 shows Calamari performed well. But the comparison was limited only to the line by line conversion, as Calamari only supported this segmentation mode. Tesseract came up in the top 3 engines and has good community support. Tesseract is a free and open-source OCR engine and has a python as well as C++ bindings. It was initially developed in Hewlett Packard Laboratories Bristol between 1985 and 1994. And was later introduced in 1995 at UNLV Annual Test OCR accuracy [14]. The development is supported by Google from 2006 onwards and now includes 116 different language data.

Early work on the construction of the face tracking robot was published where fundamental performance measuring testbed was made [15]. Here, we proposed a testbed for comparing the performance of face tracking mechanism, aided with both hardware and software support. We experiment and choose the best algorithm for face detection, recognition and tracking on a hardware limited platform. The mechanisms and simulations for the movement of the robotic arm, face detection, identification and other visual capabilities implemented on the testbed with their experimental results are discussed in this work. As more features are getting added to the testbed, it can be considered as a fully functional interactive robotic testbed for assessing the performance of various machine learning methods on hardware constrained embedded environment.

2. DESIGN AND CONSTRUCTION OF ROBOTIC TESTBED

The proposed system architecture of Interactive Robotic Testbed includes a camera which feeds the captured images into the processing part as shown in Fig. 1. In the processing part, the frames are analysed to detect faces and gestures. The main goal of the hardware system is to track the position of human faces in the real world. Based on the results, the trajectory of motion is planned using inverse kinematics.

Kinematic and Dynamics library (KDL) was used for trajectory planning of the test bot. The KDL is used for the 3D frame and vector transformations, as well as to solve Kinematics and Dynamics of kinetic chains. Kinetic chains are the relationship between links and joints. The kinetic chains are described by Unified Robot Description Format(URDF) and is further converted to KDL tree by using KDL parser. Flexible links can't be represented using URDF format, but our proposed structure contains Rigid links connected with joints.

Inverse kinematics is used to calculate the angles $\theta 1 - \theta 5$ corresponding to each servo from the end-effector position. KDL is used for calculating the inverse kinematics of the robot from the Transformation matrix. Based on the values from face detection block the KDL will calculate the unknown θs. The robot was simulated using petercorke's robotic toolbox for MATLAB as in Fig. 2 (a) and using RViz visualization tool as in Fig. 2 (b).

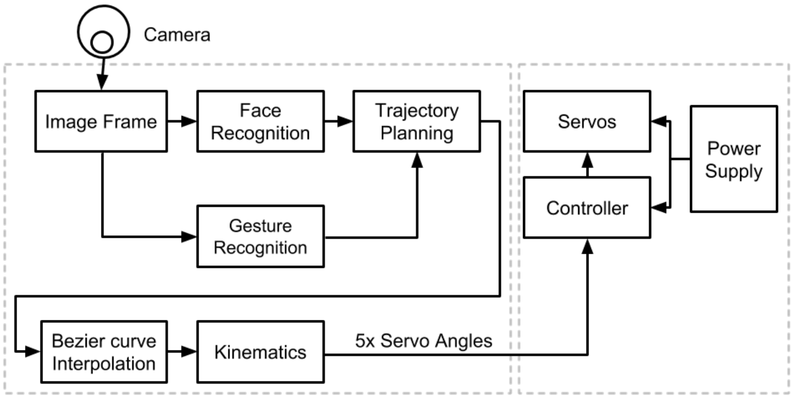
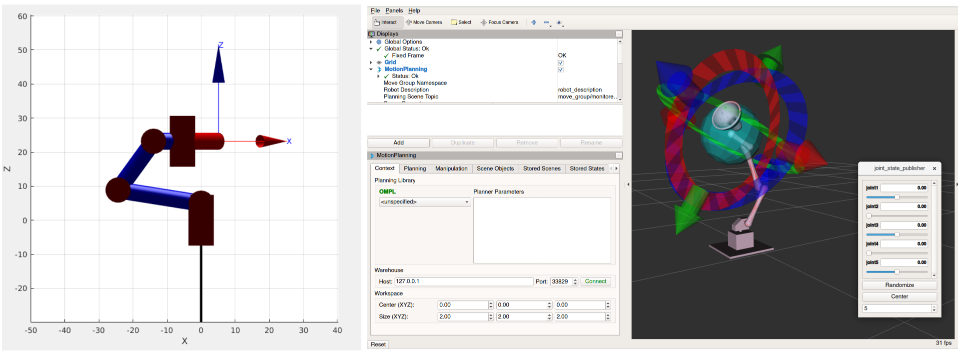


Fig. 1. System architecture of robotic testbed.



(a) MATLAB output.

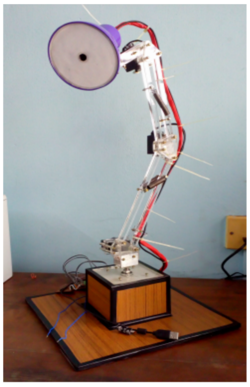
(b) RViz output.

Fig. 2. Simulations.

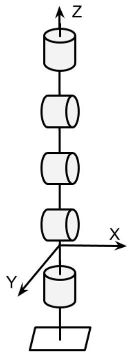
Whenever the robot runs short of queued points to be approached, it invokes a pattern point generator, which is parameterized with the current pattern. There, the next point of a predefined sequence of movements is generated. Bezier curve interpolation is used to interpolate the trajectory for a smoother movement and keeps the robot acceleration stable.

The joint rotation diagram of the robot is given in Fig. 3 (b). All of the joints in the robot are revolute type. The revolute joint provides single-axis rotation. Each revolute joint has one-degree-of-freedom. The robot consists of 5 revolute joints and thus have 5 degrees of freedom (DOF). The servo motors are adjusted with the help of DOF manipulator hardware.

The DOF manipulator hardware structure of the robot is as shown in Fig. 3 (a). Single board computer, Arduino Uno controls the entire system and a Switch Mode Power Supply (SMPS) provides the required power for the entire system. Arduino Uno, an ATmega328P microcontroller board, has 14 digital input/output pins of which 6 of the PWM outputs are used for rotating the servo motors and control the brightness of LEDs.



(a) Hardware structure.



(b) Joint rotation diagram.

Fig. 3. 5 DOF manipulator.

3. IMPLEMENTING MACHINE LEARNING ALGORITHMS ON TESTBED

Being conceptualized as a face tracking robot, we added basic functionality to detect, recognize and track faces. Additionally, various visual recognition capabilities were also included. This section discusses how we arrived at the best algorithms for our testbed in each of these areas.

3.1 Face Detection

Humans differentiate objects utilizing physical attributes. However, physical attributes are rarely a commendable choice for computers. Different feature extraction methods like Haar features and LBP features [16] are widely used to detect objects in images. The Haar feature extractor gives much more accurate detection rate, whereas the LBP feature extractor is 10-20% less accurate. LBP extractor has the advantage of better speed than Haar at the cost of accuracy. An improved recognizer can be implemented by using deep learning algorithms.

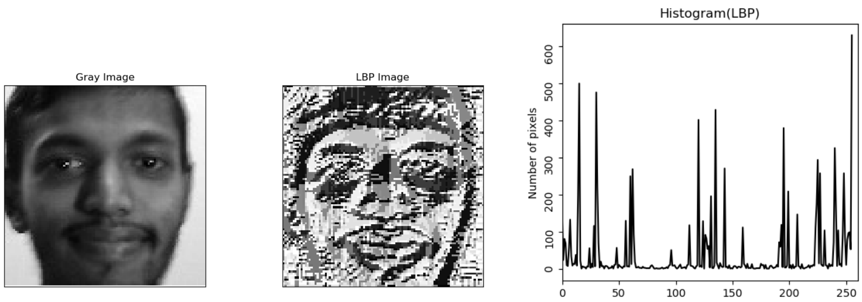
Face detection by Deep neural networks (DNN) is implemented with Caffe modules and OpenCV. Different types of models are there which varies by speed and accuracy depending on the model size and the logic used. Faster RCNN has good accuracy at the sacrifice of speed. Resnet models are larger models but perform exceptionally well on most use cases. SSD models have sufficient speed and relatively good accuracy too. Considering the speed vs accuracy tradeoff, we decided to use SSD with Resnet as base architecture.

The dataset was prepared by ourself containing about 5000 facial images. Dataset is split into training and validation set in a 3:1 ratio. To feed the Caffe models with such large dataset, LMDB data format is commonly used.

In our work, we implemented Haar, LBP and Deep Neural Networks on the testbed and assessed their performance. The results and analysis are given in Section 5.

3.2 Face Recognition

Face recognition is performed by using Local Binary Pattern Histogram (LBPH) recognizer [17] in OpenCV. Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighbourhood of each pixel and considers the result as a binary number [18]. A detailed writeup on how LBP works is given in [15]. The output of circular LBP operation and the generated histogram is given in Figs. 4 (a) and (b) respectively.



(a) Thresholding operation on an image. (b) Histogram.

Fig. 4. Circular LBP and histogram.

Each histogram created from real-time images is compared with an equivalent histogram of images in the training dataset. There are various metrics to compare histograms such as Chi-Square, Euclidean Distance and Absolute Value. In our implementation, we used the Euclidean distance(1) based metric to provide confidence level in face matching.

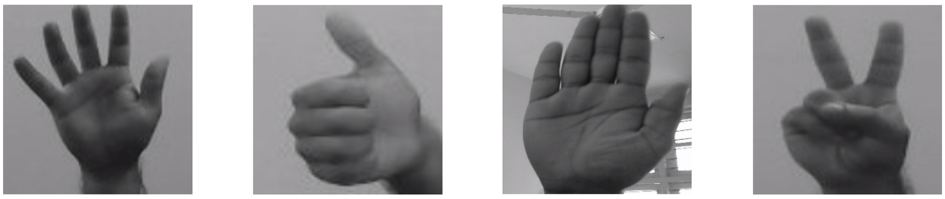
$$D = \sqrt{\sum_{i=1}^{n} (hist1_i - hist2_i)^2} \tag{1}$$

A unique identification number (ID) from the image with the closest histogram in the dataset is taken. It also returns the calculated distance, which can be used for confidence measurement. Based on this confidence level, the robot decides if a person is matched to the face in the dataset. For a recognized person, customized interactions and special movements are incorporated.

3.3 Adding Visual Intelligence

As a face tracking bot, it has great potential in areas such as surveillance and security. Adding visual intelligence capabilities enhances the bot further. Gestures play an important role in making the Audio-Visual communication more effective and meaningful. Humans do this a lot by hands, facial expressions or even a simple whole body posture. Similarly, to make the Human-Machine Interaction effective and meaningful, it is necessary to add gesture recognition to our interactive robotic testbed. We used five different gestures to convey commands such as sleep, wake up, switch faces, read and to control a light bulb connected on the robot.

The challenge is to extract the hand position properly from the background in real-time. As we already have face detection and tracking system implemented, the camera will always be moving. This will make further processing difficult due to motion blur in the images. The simplest way to detect hands are based on skin colour, but it can change with different people as well as the chances of detecting other objects as false positives increases. Skin colour detection may detect the user's arms or faces instead of hands, making this method unreliable.



Change Face Sleep Toggle LED Wakeup

Fig. 5. Samples from gesture image database.

Initially, we collected images with different hand gestures for various commands as in Fig. 5. We then experimented with various cascades in OpenCV to train and detect the hands. We found that the best performance is when applying local binary patterns operator and LBPH is used to describe the hand image. Local binary patterns operator is a texture descriptor which transforms an image into a histogram of codes, where each bin corresponds to the number of similar patterns describing texture patches. Inspired by the table lamp structure of our robotic testbed, we tried to make the testbed capable of reading and converting image frames that contain text to speech. An OCR engine analyses the frames and extracts the text. A speech synthesizer converts the text into sound and plays back to the user. Once the reading mode on the testbed is activated, the robotic arm is set to a predefined position and captures the image after a delay to avoid motion blur.

4. EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Comparative Study: Face Detection Algorithms

Face detection is a mandatory process before tracking. In the evaluation of the proposed work, the faces were detected and tracked correctly. For all tests lighting conditions were stable. Performance of LBP, Haar feature extractors are compared with the Caffe framework based Deep Neural Network to find the best face detection method. The comparison results are given in Table 1.

From the comparison, the DNN based face detection has better accuracy (97.5%) than other feature extractors. However, LBP has lower memory usage, latency and processing delay when compared with Haar and Caffe DNN models. One of the issue with the Caffe DNN is the low throughput (18 frames). But for our application, throughput should be sufficient enough and more importantly needs better accuracy. Multiple tests by us have shown that face detection can cope up with conditions such as wearing glasses, using masks, and some orientation of the head. Detection results are presented in Fig. 6.

Table 1. Performance comparison for feature extractors.

Sl.No	Parameter	*LBP*	*HAAR*	*DNN Model*
1	Accuracy(%)	63	86.17	97.52
2	Memory Usage (MB)	57.13	60.94	91.42
3	Latency (ms)	586.57	600.45	676.61
4	Processing Delay (ms)	13	25	55.5
5	Throughput (frames)	76.9	40	18
6	CPU Utilisation (%)	35.5	67.3	37.25

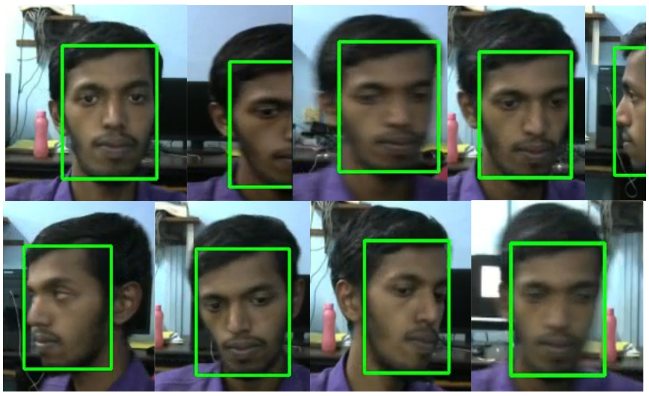


Fig. 6. Face detection result.

We found that the proposed system fails in the face detection part if eyebrows and both eyes are not visible. Faces in extreme positions also make it difficult for the system to successfully detect them. One of the reasons for the failure of face detection is the non-detection of complete eye features.

Once the face is detected, we use LBPH algorithm to identify the user. As the LBPH algorithm creates a unique histogram feature for each registered user, we require the users to first enrol themselves for face recognition. The identification accuracy was up to 98% in our experiments with a limited number of registered users. But as the number of registered users increased, the accuracy dropped. Further research has to be done, to rectify this problem. User identification helped us to personalize the testbot. Customized interactions were provided for registered users.

The trajectory planning with Kinematics and Dynamics Library (KDL) was not as fast as expected. FastIK is another similar library which is found to be faster Inverse Kinematics Library for DOF less than 5. The older Linux dependencies and higher degrees of freedom refrained us from testing fastIK on the testbed.

4.2 Experimenting with Visual Capabilities

Designed as a potential surveillance face tracking bot, additional visual intelligence can truly enhance its performance. Ability to extract text and reading back to the user as well as gesture recognition are two capabilities which we tried to demonstrate. We chose the opensource python library Tesseract [19] as the OCR engine for our testbed.

The Tesseract neural network module, based on Long Short Term Memory (LSTM), has higher accuracy compared to traditional tesseract library. As we are currently focusing only on the English language, we can use pre-trained models and can be fine-tuned as necessary. For preparing the dataset, the images have to be taken in TIFF format and a corresponding text file (.gt.txt) is made with actual content in the images.

In our experiments, the OCR works as expected on high-quality images and properly aligned texts. Images taken from the testbed had to be pre-processed before giving to the OCR for text recognition. Hardware, as well as software limitations of the testbed, has a great impact on the bot's performance. As we are having a moving testbed with 5 degrees of freedom, the chances of motion blur in the images taken are high. The text may not be properly oriented with the bot's camera or position. The tilted text, camera quality and lighting conditions are major factors affecting OCR performance.

The accuracy of the OCR engine can be improved by image pre-processing to make sure the image has properly aligned texts. Skew up to 5 degrees is corrected by Tesseract itself. Proper scaling to at least 20-pixel height, rotation or skew more than 5 degrees have to be corrected in the pre-processing pipeline, before giving to OCR. For re-alignment, we first estimate the skew of the image like in Fig. 7 (a) and then rotate the image to de-skew it. First, find all the coordinates of the foreground text and make a bounding box which contains all the data. Estimate the angle at which this box is rotated. The image is rotated to this angle to get the text realigned as in Fig. 7 (b).

However, In our experiments, we found that there is some ambiguity between certain numerical and alphabetical characters. Eg: O and 0, 2 and Z, 5 and S, 4 and A, 8 and B. Also punctuations like $(.,-_;:)$ were difficult to detect. We can use regular expressions to parse known fields such as dates. Google Text to Speech (GTTS) library is used to convert the recognised text to audio format.

(a) Skewed test image. (b) Corrected image for OCR. (c) Gesture detection result.

Fig. 7. Visual capabilities.

For detecting gestures, We collected positive images of about 1000 images for each of the five hand gestures that we intend to train. Another 1000 images were also taken, of which we need to avoid being recognized. Before training, the images were pre-processed, converted to grayscale, resized to 300*300 images and are written to a vector file. After training with 15 cascade stages, the normalized LBPH of the training image dataset is written to an XML file. Instead of running a single recognizer, we ran 5 parallel detectors to increase the efficiency of the system. The gesture recognised is based on the output of all five detectors. The Euclidean distance was calculated and the smaller its value, the more similar is the new image to that of the image in the database. A confidence score of 80% is set as a threshold for the detection.

We trained the hand recognition detector separately from the testbed and was able to run the detector at the standard 30 frames per second. But on integrating to the system, we got only 18 fps with 80%accuracy, as the face detection DNN model provides 18 frames throughput. Higher accuracy is seen in better light conditions and larger resolutions. The gesture recognition result is shown in Fig. 7 (c).

5. CONCLUSIONS

We have proposed and developed a real-time interactive robot as a testbed for assessing the performance of various computer vision techniques which are powered by current machine learning advancements. In the proposed system, the robot can detect, identify and track a face in real-time. The camera is placed on a robotic arm with five degrees of freedom. The hardware support can help the testbot to cope up with occlusions, intended or unintended, which can make the tracker lose the tracking object normally.

Considering the limited robotic/embedded hardware environments, the best performing face detection algorithm has been found by comparing various face detection techniques. The bot acted as a testbed to compare various face detection algorithms like Haar, LBP and Deep Neural Networks. We did expect some biases in age, gender, ethnicity, facial accessories due to our limited dataset. Facial identification feature was achieved using LBPH based face recognition algorithm. The detection, identification and tracking results were accurate and have real-time performance (up to 18 frames per second). But the system is found to fail under poor lighting conditions.

Adding visual intelligence to the hardware based face tracking system was a challenging task. Resouce consuming face analysis algorithms made it difficult to add additonal capabilities to the bot. For the gesture recognition system, which can recognise up to five different hand gestures, speed was prefered over accuracy. For a different circumstance and application, preference would be different. LBPH was our choice for detecting hand gestures and got 80% accuracy even under low light and lower resolutions. The additional integration did not eventually slow down the overall performance of the system and continues to work at 18 fps. Additionally, we added more visual intelligence with text character recognition from the image. Based on multiple peer studies, we decided to implement the Tesseract Engine, an open-source OCR framework. Finally, Google Text to Speech(GTTS) converts the recognised text to audio content and plays back. Images with motion blur, poor lighting and unaligned text were not recognised correctly. The OCR accuracy was improved with enhanced images and proper rotation and skew. Adding spell check, dictionary, auto-correction, regular expressions to improve the OCR result.

In conclusion, we believe approximations are necessary to keep the speed vs accuracy under required limits for a robot. Demands of each application would be different. Personalising the interaction and response of the testbot based on user sentiments are efforts being done in extending the scope of the robot. A video of the interactive testbot is in reference [20].

REFERENCES

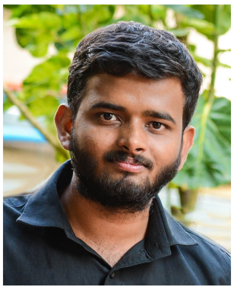
1. A. Arunmozhi and J. Park, "Comparison of hog, lbp and haar-like features for on-road vehicle detection," in *Proceedings of IEEE International Conference on Electro/Information Technology*, 2018, pp. 0362-0367.
2. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886-893.
3. K. Meena and A. Suruliandi, "Local binary patterns and its variants for face recognition," in *Proceedings of International Conference on Recent Trends in Information Technology*, 2011, pp. 782-786.
4. M. D. Putro and K.-H. Jo, "Real-time face tracking for human-robot interaction," in *Proceedings of International Conference on Information and Communication Technology Robotics*, 2018, pp. 1-4.
5. R. Boda and M. J. P. Priyadarsini, "Face detection and tracking using klt and viola jones," *ARPN Journal of Engineering and Applied Sciences*, Vol. 11, 2016, pp. 13 472-13 476.
6. K. Dang and S. Sharma, "Review and comparison of face detection algorithms," in *Proceedings of IEEE 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence*, 2017, pp. 629-633.
7. D. Lee, H. Yang, and D. Lee, "Preliminary results on vision-inertial state estimation for smqt system," in *Proceedings of IEEE 12th International Conference on Ubiquitous Robots and Ambient Intelligence*, 2015, pp. 161-162.
8. M.-H. Yang, D. Roth, and N. Ahuja, "A snow-based face detector," in *Advances in Neural Information Processing Systems*, 2000, pp. 862-868.
9. P. Viola, M. Jones, *et al.*, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, 2001, pp. 511-518.
10. M. Komar, P. Yakobchuk, V. Golovko, V. Dorosh, and A. Sachenko, "Deep neural network for image recognition based on the caffe framework," in *Proceedings of IEEE 2nd International Conference on Data Stream Mining and Processing*, 2018, pp. 102-106.
11. R. M. Gurav and P. K. Kadbe, "Real time finger tracking and contour detection for gesture recognition using opencv," in *Proceedings of IEEE International Conference on Industrial Instrumentation and Control*, 2015, pp. 974-977.
12. A. S. Ali and Y. Xiang, "Spam classification using adaptive boosting algorithm," in *Proceedings of the 6th IEEE/ACIS International Conference on Computer and Information Science*, 2007, pp. 972-976.
13. C. Wick, C. Reul, and F. Puppe, "Calamari-a high-performance tensorflow-based deep learning package for optical character recognition," *arXiv preprint arXiv:1807.02004*, 2018.
14. S. V. Rice, F. R. Jenkins, and T. A. Nartker, "The fourth annual test of ocr accuracy," Technical Report 95-03, Information Science Research Institute, University of Nevada, 1995.
15. P. Nithin, F. Albert, J. C. Ajai, A. J. Bijoy, and J. Mathew, "Face tracking robot testbed for performance assessment of machine learning techniques," in *Proceedings of IEEE*

7th International Conference on Smart Computing and Communications, 2019, pp. 1-5.

16. D. T. P. Hapsari, C. G. Berliana, P. Winda, and M. A. Soeleman, “Face detection using haar cascade in difference illumination,” in *Proceedings of IEEE International Seminar on Application for Technology of Information and Communication*, 2018, pp. 555-559.
17. X. Zhao and C. Wei, “A real-time face recognition system based on the improved lbph algorithm,” in *Proceedings of IEEE 2nd International Conference on Signal and Image Processing*, 2017, pp. 72-76.
18. C. Chai and Y. Wang, “Face detection based on extended haar-like features,” in *Proceedings of IEEE 2nd International Conference on Mechanical and Electronics Engineering*, Vol. 1, 2010, pp. V1-442.
19. R. Smith, “An overview of the tesseract ocr engine,”in *Proceedings of IEEE 9th International Conference on Document Analysis and Recognition*, Vol. 2, 2007, pp. 629-633.
20. B. A. Jose, “Face tracking ai robot – machine learning,” Department of Electronics, CUSAT, https://www.youtube.com/watch?v=rU5locu6Mbg.

Nithin P. B. has completed his Post Graduation in Electronics Science from Department of Electronics, CUSAT, Kerala, India. He obtained his B.Sc degree from N.S.S College Rajakumari, Kerala, India. He is currently working as a trainee in Pumex Infotech Pvt. Ltd. His areas of interest include artificial intelligence, machine learning, penetration testing.

Albert Francis Rathappillil has completed his M.S. in Electronics and Communication from Department of Electronics, CUSAT, Kerala, India. He is specialised in embedded systems and their security. He is currently working as a Research and Development Engineer in Marine Power Electronic Systems. His areas of interest include embedded systems, wireless communication and power electronics.



Ajai John Chemmanam currently a Research Scholar in Artificial Intelligence and Machine Learning at Department of Electronics, CUSAT, Kerala, India. He obtained his Master and Bachelor degrees from Cochin University of Science and Technology, Kerala, India. He has two years of industrial experience in AI and ML. His research interests include security threats of neural networks, image and video analytics.

Bijoy A. Jose currently working as an Assistant Professor in Department of Electronics, CUSAT, Kerala, India. He has received his B.Tech from School of Engineering CUSAT and MS from State University of New York. He has been an intern with IBM, New York and Intel Corporation, Illinois during his graduate studies. He received his Ph.D. from Virginia Tech and worked in Intel Corporation in California and Bangalore for four years. He has received Early Career Research Award from Department of Science and Technology, Govt. of India in 2016. He is Principal Investigator to multiple funded projects from DST and IEEE. His areas of interests include cyber security, Internet of Things and cyber physical systems.

Jimson Mathew currently an Associate Professor and Head of Computer Science and Engineering Department, Indian Institute of Technology, Patna, India. He received the masters degree from Nanyang Technology University and Ph.D. from University of Bristol, UK. He holds multiple patents, is the co-author of three books, and more than 100 publications in international journals and conference of repute. His research interests include fault-tolerant computing, hardware security and machine learning architectures