International Journal on Emerging Technologies **11**(3): 313-319(2020)

ISSN No. (Print): 0975-8364
ISSN No. (Online): 2249-3255

Qualified Scrutiny for Real-Time Object Tracking Framework

Ravindra R. Patil[1], Omkar S. Vaidya[2], Gayatri M. Phade[3] and Sanjay T. Gandhe[4]
[1]Post Graduation Scholar, Department of Electronics & Telecom. Engineering,
Sandip Institute of Technology & Research Centre (Savitribai Phule Pune University), Nashik (Maharashtra), India.
[2]Assistant Professor, Department of Electronics & Telecom. Engineering,
Sandip Institute of Technology & Research Centre (Savitribai Phule Pune University), Nashik (Maharashtra), India.
[3]Associate Professor and Head, Department of Electronics & Telecom. Engineering,
Sandip Institute of Technology & Research Centre (Savitribai Phule Pune University), Nashik (Maharashtra), India.
[4]Professor and Principal, Department of Electronics & Telecom. Engineering,
Sandip Institute of Technology & Research Centre (Savitribai Phule Pune University), Nashik (Maharashtra), India.

(Corresponding author: Ravindra R. Patil)
(Received 30 January 2020, Revised 03 April 2020, Accepted 06 April 2020)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: In this modern edge of the world, life is turning out to be more and more palatial due to resourceful technology but security is an overriding concern. The tracking offered significant robustness to the security approaches. The discipline generally based on interest such as pedestrian tracking, vehicle tracking, and surveillance systems, etc. This paper represents qualified scrutiny for tracking API's in OpenCV such as BOOSTING, TLD, MEDIANFLOW, MIL, MOSSE, CSRT, KCF, and GOTURN for speed and accuracy. The newer cost effective, powerful Raspberry Pi 3 Model B+ and another Intel ULV Processor with lofty specifications are two different systems used for running OpenCV and webcam for capturing real-time video frame for further analysis. It is really tricky and challenging part to work on these two distinct systems and different natured trackers for accomplishing a task of tracking for intended consequences of visual upshots and computations which highly differs in many technical aspects. The python language is used for effective coding due to its simple syntax with multi-threading in OpenCV. From experimental results, the qualified scrutiny has evidenced or substantiated that which tracker will be pre-eminent for real-time tracking framework in the practical world according to essentials.

Keywords: Intel ULV Processor, Object Tracking, OpenCV, Python, Qualified Scrutiny, Raspberry Pi 3 Model B+.

Abbreviations: OpenCV, Open source Computer Vision Library; TLD, Tracking-Learning-Detection; MIL, Online Multiple Instance Learning; MOSSE, Minimum Output Sum of Squared Error; CSRT, Channel and Spatial Reliability; KCF, Kernelized Correlation Filter; GOTURN, Generic Object Tracking Using Regression Networks; ULV Processor, Ultra Low Voltage Processor; FPS, Frames Per Seconds.

I. INTRODUCTION

Today, human day to day life and the distinct working fields are full of unlike objects with conflicting identities. In such a state, security is getting to be predominant affair and object tracking is a major sense of it. The autonomous vehicle navigation, medical imaging, warehouses, traffic control systems, schools or college security, home and social security are some instances where tracking is a crucial strand of them and are varying unceasingly.

Object tracking is locating an object in successive video frames. It consists of the proposals like kalman filtering, dense optical flow, meanshift and camshift, single and multi- object trackers and sparse optical flow etc. The eight object tracking algorithms are accessible in OpenCV like BOOSTING, TLD, MEDIANFLOW, MIL, MOSSE, CSRT, KCF, and GOTURN. These are well known single object tracking algorithms in OpenCV and all are available in 3.4+ versions. From the above trackers, only GOTURN is deep learning based object tracker for realizing real-time adequate results. Here, Raspberry Pi 3 model B+ single board computer is used for provisional scrutiny of trackers in OpenCV with reference to speed and accuracy for real-time cost effective tracking frameworks. Similarly, qualified scrutiny has been petitioned for object trackers on Intel ULV Processor carrying lofty detailing. For coding, python is exploited and webcam is used for real-time video streaming.

All the related work for object trackers given in section II enlarges the object tracking platform but it's hard to decide the proficiency in practice and real world implementations though come across all concepts and theories. So, we perform the qualified scrutiny for these trackers to unfold every bit of key features such as speed and accuracy for practical implementations according to need in respective domains.

The paper is conferred into six sections. The Section II provides the literature survey and Section III unfolds the mode of operation. The experimentation of the system is explicated in the Section IV whereas the results and skull session are proffered in Section V. After all, the conclusion of the paper is registered in Section VI.

II. LITERATURE SURVEY

Vaidya *et al.*, (2019) focused on embedded vision in which cost effective tele-operating smart robot has been developed. The proposed system is capable to identify the person using LBPH face recognizer, employing eSpeakNG for speaking the name and human tracking is consummated smartly with the particularity of shunning obstacle by making use of HOG with linear SVM, non-maximum suppression to remove excessive overlapping bounding boxes during human detection and KCF for tracking. For intercommunication, Alexa Voice Service has been enabled. This robot can be positioned for distinct fields of security [1].

Grabner and Bischof (2006) inaugurated futuristic boosting based feature selection online algorithm which uses HAAR classifier internally. The positive and negative samples of the object are utilized for training purpose of the classifier. The user or object detection algorithm given bounding box is considered to be a positive sample and remained is a backdrop. The classifier score for each new frame is noted and where the maximum score found is a new location of the object. The classifier is upgraded with new data progressively [2].

Kalal *et al*., (2010) came up with an innovative method of tracking failure detection based on the Forward-Backward error and gave the name as “Median Flow”. According to the study, forward and backward object tracking is done by the tracker in time and both trajectories differences are calculated. The tracking failures can be detected effectively by reducing Forward-Backward error. It also gives a contribution to the selection of reliable trajectories in video sequences [3]. Kalal *et al*., (2012) presented the next work on a “TLD” tracker for long-term tracking of anonymous objects in a video stream. TLD intends Tracking, Learning and Detection mean the task is deteriorated into these keywords. The detectors isolate all the detected aspects till that movement and rectify the tracker if imperative. The Learning guesses the detector’s error and renovates it to circumvent errors in forthcoming [4]. Several other multiple object detection techniques reviewed in [5-8] which address issues of cluttered scene, occlusion and undesired transformations.

Bolme *et al.,* (2010) proposed the new algorithm for real-time object tracking known as “Online Multiple Instance Learning”. It uses the idea alike to the boosting tracker and considers not only the current location but also tiny region around it with the specification of positive and negative “bags”. Overall this paper focused on the problem of learning an adaptive appearance model for object tracking [9]. Babenko *et al.,* (2009) caught the attention to the modern kind of correlation filter, a “Minimum Output Sum of Squared Error” (MOSSE) filter. When it is initialized by utilizing a single frame, processes stable correlation filters. This algorithm is tough to the conditions such as light variation, scale, pose, and non-rigid deformations and also capable to detect occlusion [10].

Lukežic *et al.,* (2017) submitted a perspective on discriminative correlation filter which uses two standard features as HoGs and Colornames. The filter support to the object part is acceptable for tracking which is adjusted by the spatial reliability map. Then, enlargement of search region and improvement in tracking of non-rectangular objects are assured. The CSR-DCF method attained first-rate results on VOT 2016, VOT 2015 and OTB100 [11]. Henriques *et al*., (2015) introduced “Kernelized Correlation Filter” (KCF) for tracking which has been formed on the recommendations given in MIL and BOOSTING trackers. Here, the cordial mathematical properties caused by overlapping data utilized by KCF tracker for faster and more accurate tracking [12].

Held *et al*., presented a technique for online training of neural networks and the tracker based on it is GOTURN which stands for “Generic Object Tracking Using Regression Networks”. The tracker utilizes a simple feed-forward network and learns a generic relationship between object motion and appearance which further used for tracking. The GOTURN originally developed in Caffe and then after interfaced to tracking API in OpenCV. It is capable to track objects at test-time at 100 fps [13]. The OpenCV Tracking API Classes (available online) have all eight tracking algorithms viz. BOOSTING, TLD, MEDIANFLOW, MIL, MOSSE, CSRT, KCF, and GOTURN [14].

III. MODE OF OPERATION

The proposed system has focused on qualified scrutiny for OpenCV tracking API’s and these trackers already discussed in section II. The OpenCV is an open source computer vision and machine learning library with a robust computational capability resorted for image processing in a real-time application. The Raspberry Pi 3 Model B+ and Intel ULV Processor are used for running OpenCV. The python is a powerful open source language used for efficacious coding on both hardware systems. Fig. 1 shows the block diagram of the proposed work.

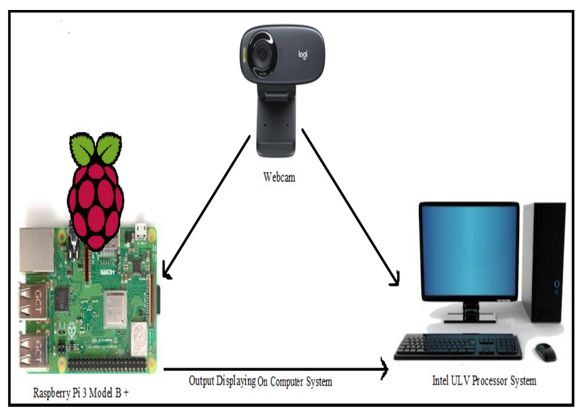

Fig. 1. Block Diagram of Proposed System.

The Raspberry Pi 3 Model B+ comprehends Broadcom BCM2837B0 with Cortex-A53 (ARMv8) 64-bit SoC at a 1.4GHz processor. The 2.4 GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN and Bluetooth 4.2, BLE are for wireless interconnections. The 40-pin GPIO header for connecting input-output components and 1GB LPDDR2 SDRAM involved in it. The parallel computing of video data is supported by Broadcom Videocore-IV GPU at low clock speed [15]. It can be a brilliant alternative for initializing cost-effective security system. Also, Intel ULV Processor has been utilized which involves detailing such as Intel(R) Core(TM) i5-3337U CPU at 1.80GHz and Intel(R) HD Graphics 4000. The mechanism is begun with capturing a real-time videostream by Logitech webcam and processed in Raspberry Pi for the desired tracking. The provoked outcomes are shown on the computer system. The same strategy has been used for Intel ULV Processor which processes the video stream from a webcam and shows results on the monitor screen. The multithreading used in coding made video processing conduit additionally faster which actually influenced real-time object tracking. The "cv2.selectROI" function in OpenCV gives the access to the user to select the region of interest on a frozen videostream frame with a mouse by popping intended key that we want to track as shown in Fig. 2.

Here, the cellular phone, face, and ball have been stipulated by rendering bounding box for frontward functioning. Also different objects of a kind toy car, eye, human, pictures on the books, contrasting flower, etc. are used for analysis.

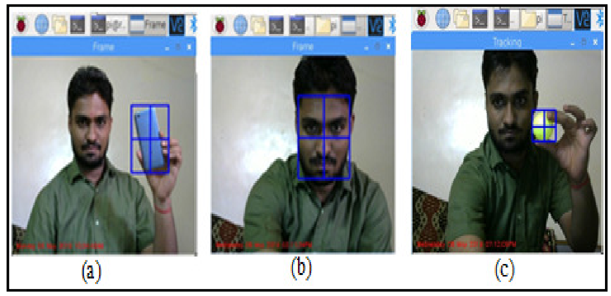

Fig. 2. Selected cellular phone in frame (a), face in frame (b) and ball in frame (c) for tracking aspiration.

Without a doubt, an indubitable, palpable object detector can be used instead of hand-operated assortment.

IV. EXPERIMENTATION

All eight tracking algorithms have been qualified in the sense of speed and accuracy. The Raspberry Pi 3 model B+ is a robust and pocket worth processor for innovative and integral advancement that is the reason for citation. Also, Intel ULV Processor is also cogitated with substantial specifications as shown in Table 1.

.**Table 1: Comparison between Raspberry Pi and Intel Ultra Low Voltage (ULV) Processor.**

System Type	Raspberry Pi 3 Model B+	Intel ULV Processor
Processor	Broadcom BCM2837B0 with Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz CPU	Intel(R) with Core(TM) i5-3337U CPU @ 1.80GHz
Operating System (OS)	Raspbian Stretch	Windows 10
RAM	1 GB	4 GB

Table 2: Real-Time FPS Outcomes on Raspberry Pi 3 Model B+.

Tracker Type	Served Time in seconds	Frames per seconds
BOOSTING	2.4667	2.84
TLD	4.2543	2.57
MEDIANFLOW	3.4894	13.39
MIL	3.8451	1.29
MOSSE	4.2146	18.29
CSRT	3.2598	3.67
KCF	2.8954	7.90
GOTURN	3.9467	0.29

Table 3: Real-Time FPS Outcomes on Intel ULV Processor.

Tracker Type	Served Time in seconds	Frames per seconds
BOOSTING	3.8993	16.13
TLD	4.7181	8.72
MEDIANFLOW	2.6784	98.76
MIL	3.4297	12.81
MOSSE	2.2061	143.13
CSRT	4.6421	13.94
KCF	3.9301	84.31
GOTURN	6.3394	4.98

Here, Table 2 and 3 accommodate the details related to the frames per seconds (FPS) i.e. speed of tracking algorithms for served time on Raspberry Pi 3 Model B+ and Intel ULV Processor respectively.

The FPS is reckoned as the complete number of reading frames throughout the tracking process divided by the complete number of seconds of served time in the course of tracking.

The FPS counting is started only after continuing the frozen videostream frame of selected object for tracking intension. The resolution taken is (400x300) on both set-ups.

The Fig. 3-11 specified the real-time tracking visual upshots during experimentation on the Raspberry Pi 3 Model B+ for all eight OpenCV tracking algorithms. These tracking visual upshots depict the demeanor of all eight trackers in real-time environment.

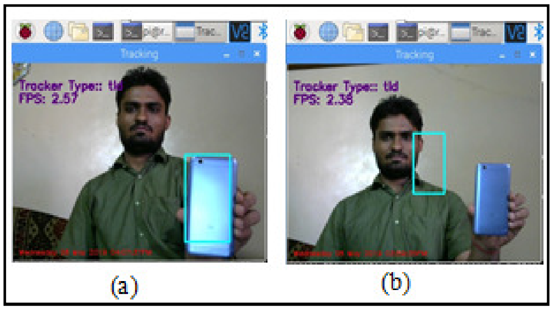

Fig. 3. Outcomes of tracking cellular phone in frames (a) and (b) with TLD tracker.

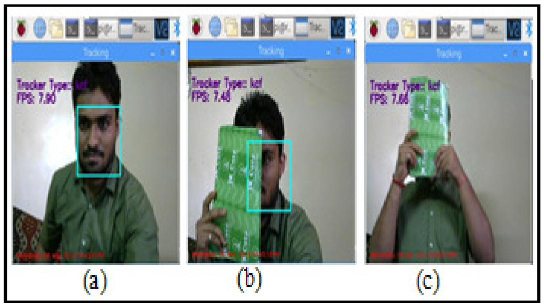

Fig. 4. Tracking face results in frames (a), (b) and (c) with KCF tracker.

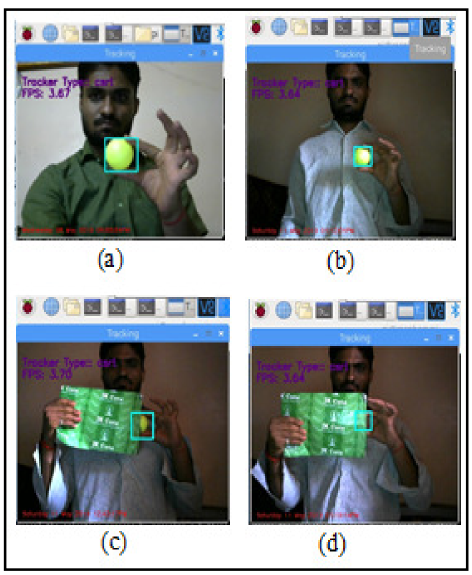

Fig. 5. Tracking results of ball in given frames (a), (b), (c) and (d) with CSRT tracker at great accuracy.

Fig. 6. Tracking book in frame (a) and rose flower in frame (b) with MOSSE Tracker at maximum speed.

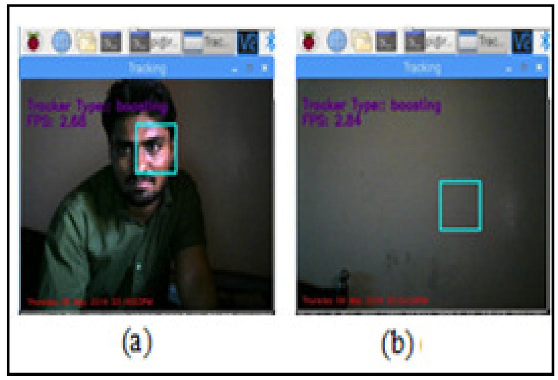

Fig. 7. BOOSTING tracker results for face tracking in frames (a) and (b).

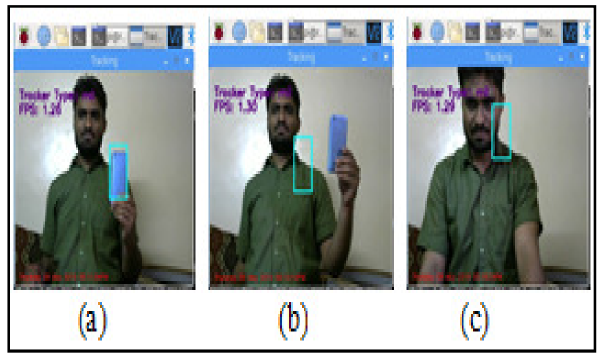

Fig. 8. Tracking results for cellular phone in frames (a), (b) and (c) with MIL Tracker.

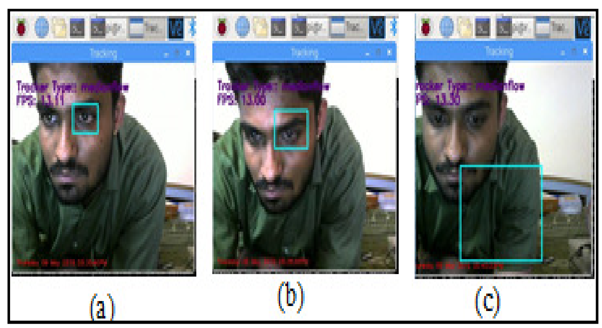

Fig. 9. MEDIANFLOW tracker results for eye tracking in frames (a), (b) and (c).

Fig. 10. Face tracking outcome in given frame by GOTURN tracker.

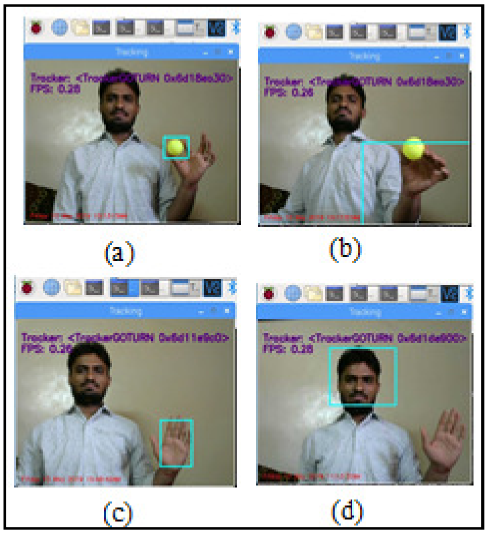

Fig. 11. Consequences for tracking ball in frames (a), (b) and palm in frames (c), (d) by GOTURN tracker.

In Table 4, the number of correct and incorrect matches for the selected region of interest during tracking is evaluated and broadcast the number of true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN). Over here, true positives (TP) connote number of correct matches, false positives (FP) indicate incorrect proposed matches, and false negatives (FN) signify not correctly detected matches and true negatives (TN) suggest correctly rejected non-matches in the course of tracking.

Table 5 involves the terms such as True Positive Rate (TPR), False Positive Rate (FPR), Positive Predictive Value (PPV) and Accuracy (ACC) which have been calculated using below equation from (1) to (4) for all eight trackers.

True Positive Rate (TPR) = TP / (TP + FN) (1)

False Positive Rate (FPR) = FP / (FP + TN) (2)

Positive Predictive Value (PPV) = TP / (TP + FP) (3)

Accuracy (ACC) = (TP + TN) / (TP + FP + FN + TN) (4)

Table 4: Contingency Table for Different Trackers.

Tracker Type	True Positive (TP)	False Positive (FP)	False Negative (FN)	True Negative (TN)
BOOSTING	15	8	5	2
TLD	9	15	2	4
MEDIANFLOW	19	4	3	4
MIL	16	7	4	3
MOSSE	20	4	2	4
CSRT	21	2	1	6
KCF	20	3	2	5
GOTURN	17	6	2	5

Table 5: Performance Parameters for Matching Strategy.

Tracker Type	True Positive Rate (TPR)	False Positive Rate (FPR)	Positive Predictive Value (PPV)	Accuracy in Percentage (ACC)
BOOSTING	0.75	0.80	0.6522	56.67
TLD	0.8181	0.7894	0.3750	43.33
MEDIANFLOW	0.8636	0.50	0.8261	76.67
MIL	0.80	0.70	0.6956	63.33
MOSSE	0.9090	0.50	0.8333	80
CSRT	0.9545	0.25	0.9130	90
KCF	0.9090	0.3750	0.8695	83.33
GOTURN	0.8947	0.5454	0.7391	73.33

V. RESULTS AND SKULL SESSION

A. Speed of Tracking Algorithms

While revolving around the real-time tracking framework, the speed and accuracy are salient vistas. Further, these attributes are discussed from acquired appraisal.

From the experimentation and from Fig. 12 and 13, it is elucidated that the MOSSE tracker is the fastest tracking algorithm which attained 18.29 FPS on Raspberry Pi 3 Model B+ and 143.13 FPS on Intel ULV Processor. In the case of deep learning based GOTURN tracker, it needs higher specifications and is a GPU oriented algorithm which gave poor outcomes on both systems such as 0.29 FPS and 4.98 FPS on Raspberry Pi 3 Model B+ and Intel ULV Processor respectively.

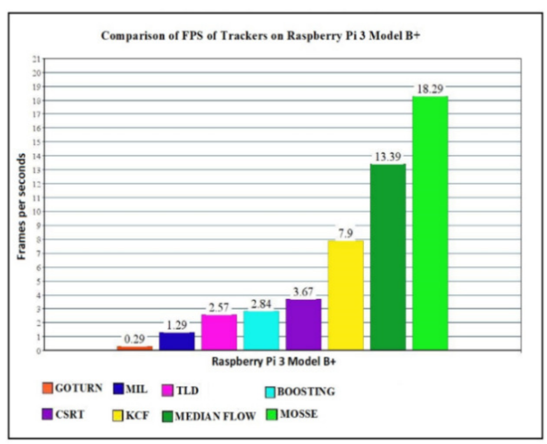

Fig. 12. Comparison of FPS of tracking algorithms on Raspberry Pi 3 Model B+.

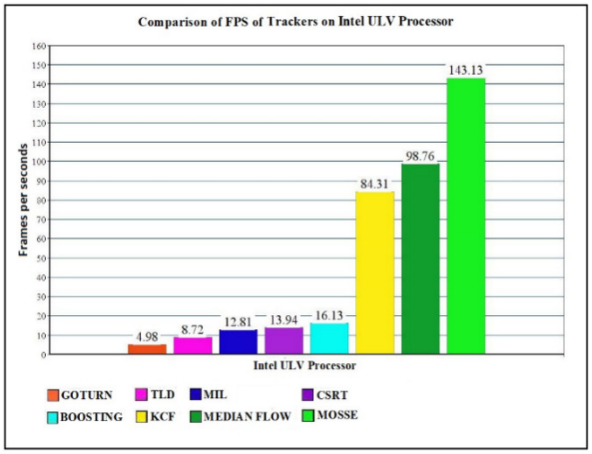

Fig. 13. Comparison of FPS of tracking algorithms on Intel ULV Processor.

The MIL tracking algorithm came to experience too slow with 1.29 FPS as well as BOOSTING, TLD and CSRT trackers also generated stagnant FPS consequences as 2.84 FPS, 2.57 FPS and 3.67 FPS respectively on Raspberry Pi 3 Model B+. On Intel ULV Processor, TLD tracker produced lesser FPS as 8.72 likewise MIL, CSRT and BOOSTING trackers are authenticated leisurely as speed of 12.81 FPS, 13.94 FPS and 16.13 FPS respectively. All in all, the KCF and MEDIANFLOW are substantiated as better in speed than other trackers but slower than MOSSE tracker on both systems. The KCF and MEDIANFLOW trackers achieved 7.9 FPS and 13.39 FPS respectively on Raspberry Pi 3 Model B+ whereas 84.31 FPS and 98.76 FPS severally on Intel ULV Processor.

B. Accuracy of Tracking Algorithms

The graph plotted in Fig. 14 substantiates the comparison of accuracy in percentage for tracking algorithms in OpenCV. The tracker CSRT with highest 90% accuracy cached an attention. This tracker plays an impressive role of tracking not only for rapidly moving object but also object in partial or full occlusion as shown in the visual upshots of Fig. 14 in the experimentations. The KCF tracker came in picture as a second higher accuracy tracker of 83.33% but does not manipulate full occlusion competently as shown in Fig. 14.

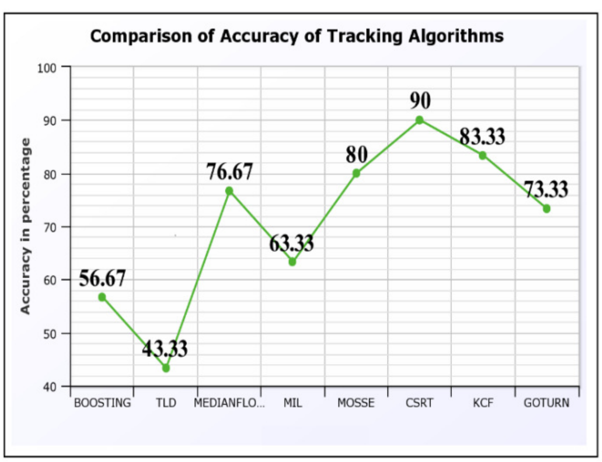

Fig. 14. Comparison of Accuracy of tacking algorithms.

The MOSSE tracker has a good 80% tracking accuracy where as for MEDIANFLOW tracker, it is 76.67% accuracy. The MEDIANFLOW tracker gave fine results for no occlusion, foreseen motion and great for describing defeat but poor to track rapidly moving objects as visible in Fig. 9. The deep learning based GOTURN tracker achieved 73.33% accuracy. This tracker sometimes tracks the object other than selected as appeared in the Fig. 11. Also tracking is performed for full object rather than selected region of that object.

The MIL tracker has better accuracy of 63.33% than BOOSTING tracker accuracy of 56.67 %. The MIL tracker also performs well in partial occlusion. From experimentation, it is clarified that the TLD tracker is practically unserviceable in real–time due to high rate of false positives with lowest 43.33% accuracy in all eight trackers.

VI. CONCLUSION

The qualified scrutiny has been accomplished for tracking API's in OpenCV such as BOOSTING, TLD, MEDIANFLOW, MIL, MOSSE, CSRT, KCF, and GOTURN dealing with speed and accuracy. It is manifested that MOSSE is the fastest tracking algorithm with speed of 18.29 FPS on Raspberry Pi 3 Model B+ and 143.33 FPS on Intel ULV Processor. While CSRT achieved highest 90% accuracy in the course of real-time tracking.

Whether there will be requirement of highest tracking accuracy and can tolerate slower FPS output i.e. speed then CSRT is the preeminent way to go. When we need admissible accuracy as well as speed then KCF is the finest alternative for real-time tracking framework which acknowledged 83.33% accuracy and speed of 7.9 FPS on Raspberry Pi 3 Model B+ and 84.31 FPS on Intel ULV Processor. If needed higher speed and can tolerate accuracy performance, then MOSSE and MEDIANFLOW trackers are an options which achieved speed of 18.39 FPS and 13.39 FPS individually on Raspberry Pi 3 Model B+ whilst 143.33 FPS and 98.76 FPS on Intel ULV Processor. The accuracy computed from experimentations for MOSSE tracking algorithm is 80% and for MEDIANFLOW algorithm 76.67%.

The deep learning based GOTURN tracker gave poor results for speed on both systems and attained 73.33% accuracy. Also MIL and BOOSTING tracker produced offensive throughputs on both systems. Due to extravagant false positives occurred in TLD tracker, it has very less accuracy as 43.33% and unacceptable for real-time tracking framework. Overall, CSRT, KCF, MOSSE and MEDIANFLOW tracking algorithms with indubitable, palpable object detector are acceptable for real-time tracking frameworks as per the requirements of user regarding to speed and accuracy in respective domain.

VII. FUTURE SCOPE

The qualified OpenCV trackers for speed and accuracy from experimentations have capability to achieve intended consequences on inexpensive, low specifications or CPU structured embedded boards for real world applications.

While considering about the Global critical situation caused by COVID-19, the human tracking and activity recognition system having low cost and higher accuracy can be originated by using qualified trackers for investigation purpose. And it will be the most significant and technical way of observing and maintaining the social distancing.

The observation and tracking system for small kids, blind person at home as well for medical care can be developed by using qualified trackers for real world execution in a pocket worth.

The autonomous navigating and tracking robotic systems for different motivations can be developed with minimum cost such as accessing in unmanned areas, in military navigation, for natural disasters, for home, society, school and college security and services.

Conflict of Interest. Authors declare that there is no conflict of interest.

REFERENCES

[1]. Vaidya Omkar, S., Patil Ravindra, Phade Gayatri, M., & Gandhe, S. T. (2019). Embedded Vision Based Cost Effective Tele-operating Smart Robot. *International Journal of Innovative Technology and Exploring Engineering, 8*(7), 1544–1550.

[2]. Grabner, H., & Bischof, H. (2006). On-line Boosting and Vision. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* New York, USA, 1362-1369.

[3]. Kalal, Z., Mikolajczyk, K., & Matas, J. (2010). Forward-Backward Error: Automatic Detection of Tracking Failures*. International Conference on Pattern Recognition (ICPR)*, Istanbul, Turkey, 2756-2759.

[4]. Kalal, Z., Mikolajczyk, K., & Matas, J. (2012). Tracking-Learning-Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 34*(7), 1409 - 1422.

[5]. Jain, S., & Chadokar, S. (2015). A Object Detection in Image Processing: A Review. *International Journal of Electrical, Electronics and Computer Engineering, 4*(2), 26-29.

[6]. Bodke, V., & Vaidya, O. (2017). Object Recognition in a Cluttered Scene using Point Feature Matching, *5*(9), 286-290.

[7]. Vaidya, O., & Gandhe, S. (2018). The Study of Preprocessing and Postprocessing Techniques of Image Stitching. *IEEE International Conference On Advances in Communication and Computing Technology, Sangamner, India,* 431-435.

[8]. Bodke, V., & Vaidya, O. (2018). Comparative Analysis of SURF and MSER based Detection Technique of Cluttered *Objects in Multiobjects Scene, 5*(6), 102-107.

[9]. Bolme, D., Beveridge J., Draper, B., & Lui, Y. (2010). Visual object tracking using adaptive correlation filters. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, USA, 2544-2550.

[10]. Babenko, B., Yang, M., & Belongie, S. (2009). Visual tracking with online Multiple Instance Learning. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* Miami, USA, pp. 983-990.

[11]. Lukežic, A., Vojír, T., Zajc, L., Matas, J. & Kristan, M. (2017). Discriminative Correlation Filter with Channel and Spatial Reliability. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* San Francisco, USA, pp. 4847-4856.

[12]. Henriques, J., Caseiro, R., Martins, P., & Batista, J. (2015). High-Speed Tracking with Kernelized Correlation Filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 37*(3), 583 - 596.

[13]. Held, D., Thrun, S., & Savarese, S. (2016). Learning to Track at 100 FPS with Deep Regression Networks. *European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands,* 749-765.

[14]. OpenCV Tracking API Classes. [Available Online At]: https://docs.opencv.org/4.0.0/d9/df8/group__tracking.html/.

[15]. Product-sheet of Raspberry Pi 3 Model B+. [Available Online At]: https://www.raspberrypi.org/products/raspberry-pi 3-model-b-plus/.

How to cite this article: Patil, R. R., Vaidya, O. S. Phade, G. M. and Gandhe, S. T. (2020). Qualified Scrutiny for Real-Time Object Tracking Framework. *International Journal on Emerging Technologies*, *11*(3): 313–319.