Penerbit Akademia Baru

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage: www.akademiabaru.com/araset.html
ISSN: 2462-1943

Tracking-learning-detection using extreme learning machine with Haar-like features

Open Access

Melisa Anak Adeh [1], Mohd Ibrahim Shapiai [1,*], Ayman Maliha [1], Muhammad Hafiz Md Zaini [1]

[1] Electronic Systems Engineering Malaysia-Japan International Institute of Technology Center of Artificial Intelligence and Robotics (CAIRO), Universiti Teknologi Malaysia, Jalan Sultan Yahya Petra, Kuala Lumpur, 54100, Malaysia

ARTICLE INFO

Article history:
Received 31 May 2016
Received in revised form 30 June 2016
Accepted 15 August 2016
Available online 19 December 2016

ABSTRACT

Nowadays, the applications of tracking moving object are commonly used in various areas especially in computer vision applications. There are many tracking algorithms have been introduced and they are divided into three groups which are generative trackers, discriminative trackers and hybrid trackers. One of the methods is Tracking-Learning-Detection (TLD) framework which is an example of the hybrid trackers where combination between the generative trackers and the discriminative trackers occur. In TLD, the detector consists of three stages which are patch variance, ensemble classifier and KNearest Neighbor classifier. In the second stage, the ensemble classifier depends on simple pixel comparison hence, it is likely fail to offer a better generalization of the appearances of the target object in the detection process. In this paper, Online-Sequential Extreme Learning Machine (OS-ELM) was used to replace the ensemble classifier in the TLD framework. Besides that, different types of Haar-like features were used for the feature extraction process instead of using raw pixel value as the features. The objectives of this study are to improve the classifier in the second stage of detector in TLD framework by using Haar-like features as an input to the classifier and to get a more generalized detector in TLD framework by using OS-ELM based detector. The results showed that the proposed method performs better in Pedestrian 1 in terms of F-measure and also offers good performance in terms of Precision in four out of six videos.

Keywords:
Tracking-learning-detection, Haar-like feature, Extreme learning machine

1. Introduction

Object tracking is an essential task within the field of computer vision where it has been used widely in medical imaging, surveillance and human computer interaction, where it can be defined to track and estimate the target in the subsequent frames [1]. In video analysis, there are three important steps involved, which are detection of targeted moving objects, tracking the objects from frame to frame and also analysis of object tracks in order to identify their behaviour. There are

* *Corresponding author.*
E-mail address: md_ibrahim@utm.my (Mohd Ibrahim Shapiai)

various applications of tracking systems such as in media production, medical application, business intelligence, robotic and arts. The task of tracking moving object is not easy as there are many challenges to be tackle such as information loss which caused by projection of the 3D world on a 2D image, noise added in images, partial or full object occlusions, complex object motion and shapes, non-rigid or articulated behaviour of objects, real time processing requirements and scene illumination changes. In order to overcome these problems, many tracking algorithms have been introduced and divided into three categories which are generative trackers, discriminative trackers and hybrid trackers.

The role of the generative trackers is to focus on how to precisely portray the object's appearance [2]. There are several advantages of the generative trackers which includes good generalization performance. When the size of training data is small, drift can be reduced if the already seen examples are reused to train the model and performance can be further improved to handle partial occlusion if the independent parts are used to represent the object. In the other hand, generative trackers tend to be confused with similar objects appear in the background when the environment is cluttered. Besides that, the trackers ignore the discriminative power with respect to the appearance of the background.

In discriminative trackers, the classifier learns the boundary between the appearance of the object and the background to maximize the difference between the object and the background or other objects [3]. The advantages of the trackers are discriminative trackers based classifiers outperforms generative trackers if sufficient training data is provided. In addition, the discriminative trackers are fast at making predictions and offer better prediction performance. The disadvantages of the trackers include sensitivity to noise hence, the discriminative trackers have to be trained with correctly labelled samples to achieve good classification performance and have less generalization performance when the training data limited which is the case when an unknown object is being tracked.

In hybrid trackers, which are the combination between the generative trackers and the discriminative trackers, the classifiers used are trained online therefore, more object appearance can be generalized and separates the object against its background or non-object region. Hybrid trackers manage to shows a good performance however, according to the theoretical discussion, an improper hybrid of discriminative and generative models generate even worse performance than pure generative or discriminative methods [4].

One example of the hybrid trackers is based on Tracking-Learning-Detection (TLD) framework where it divides the long-term object tracking into three sub-components which are tracking, learning and detection [5]. In TLD framework, the cascaded detector consists of three stages which are patch variance, ensemble classifier and K-Nearest Neighbour classifier. In the second stage, the ensemble classifier used is unable to provide a good feature value to better generalization of the appearance of the object due to its dependency on simple pixel comparison. Previously, ELM-based classifier has been employed in. In [6], the performance of the proposed approach can be improved if other features representation is used instead of using raw pixel value as this may not be able to provide better generalization of the object appearance.

In this paper, we employed Online-Sequential Extreme Learning Machine (OS-ELM) as in [6] with Haar-like features. Haar-like features are used as an input to the OSELM-based detector instead of relying on the values of a raw pixel as this features have faster calculation speed if compared to the other features. Besides that, Haar-like features are more robust against noise and lighting variation due to their dependencies.

The rest of the paper is organized as follows. In the next section, we review on the TLD framework. In section 3, we present our approach of using Haar-like feature and ELM in TLD framework. In section

4, we discuss on the methodology used. In section 5, we discuss on our finding based in the results obtained. Finally, we conclude our approach in section 5.

2. TLD framework

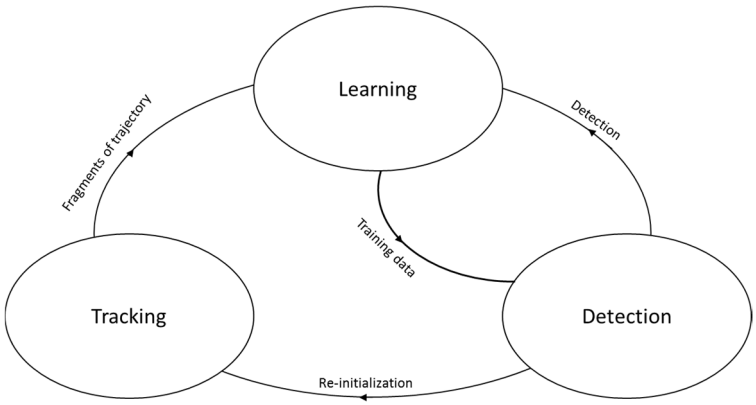

Fig. 1. TLD Framework

TLD framework [7] can be classified into three parts which are tracking, learning and detection as shown in Fig. 1. Tracking is the process of predicting next locations of previous reliable points located in the bounding box of the object and determining their reliability based on forward-backward and Normalized Cross Correlation (NCC) scores. As a result, it finds the trajectory which will be used by *P-N* experts [8]. A bounding box is used to represent the state of an object while a flag shows the object cannot be seen.

P-expert utilizes the "temporal" structure in the video and assumes that the object travels along a trajectory. The *P*-expert remembers the location of the object in the earlier frame and predicts the location of the object in present frame using a frame-to-frame tracker. If the present location is labelled as negative by the detector, a positive sample is generated by the *P*-expert. *N*-expert utilizes the spatial structure in the video and assumes that the object only appears at a single location. All responses of the detector in the present frame and the response formed by the tracker are analysed by the N-expert. It then chooses the one that is the most confident. Patches that are not overlapping with the maximally confident patch are labelled as negative sample. The maximally confident patch re-initializes the tracker location.

The function of tracker is to estimate the motion of the object between consecutive frames under the assumption that the frame-to-frame motion is limited and the object can be seen. If the object disappears from the camera's field of view, the possibility for the tracker to fail and never improve is higher.

Detection is used to find reliable bounding boxes in which the object may exist. The bounding boxes with high scores are then sent to the P-N expert for further evaluation. The aim of the detector is to determine whether the object is still in the field of camera's view and if not, the detector tries to detect the object once it comes back into the camera's view. In order to localize all appearances that have been spotted and learned in the previous experiment, every frame is treated as independent and fully scanned of the image is implemented.

Learning monitor performance of both tracker and detector, estimates the mistakes of the detector and creates positive and negative training sets to prevent the same mistakes occur in the future. The learning process is aids by the P-N experts in which the object's model is updated. The classifier is initially trained with some labelled data and then it will evaluate unlabelled data. Finally,

the P-N experts decides whether classifier's decisions on unlabelled data are correct or not by following these steps:

1) *P*-expert checks each sample labelled as negative against a trajectory. The negative is a representative of the background. If the sample is nearby the trajectory, *P*-expert re-labels the sample as positive and adds the sample to the positive training set. This will increase the generalization power of the classifier.

2) *N*-expert checks each sample labelled as positive against a trajectory. The negative is a representative of the object. If the sample is further away from the trajectory, *N*-expert re-labels the sample as negative and adds the sample to the negative training set. This will increase the discrimination power of the classifier.

If either one of the *P-N* experts or both occur, the classifier will be updated otherwise, it will remain constant as its previous condition.

3. Employed technique

3.1. Haar-like features

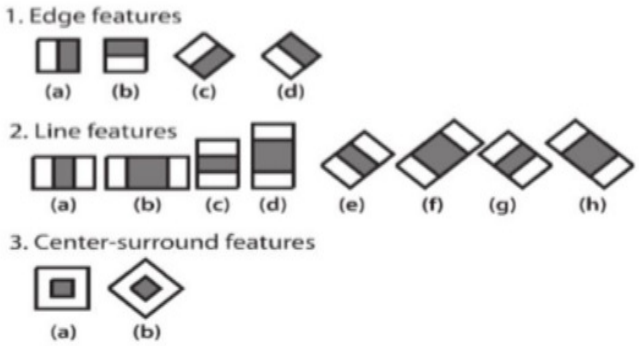

Fig. 2. Haar-like Features

Figure 2 shows, each Haar-like feature is a template of multiple connected black and white rectangles. The value of a Haar-like feature is the difference between the sums of the pixels' values within the black and white rectangular areas.

$$f(x) = W_{black}.black_{region} = W_{white}.\sum_{white\_region}(pixel\_value) \quad (1)$$

where W_{black} and W_{white} are the weights that meet the compensation condition:

$$W_{black}.black_{region} = W_{white}.white\_region \quad (2)$$

3.2. Extreme learning machine

ELM is a supervised learning technique which was originally developed for the single-hidden layer feed forward neural networks (SLFNs). The main idea of ELM is that the hidden node parameters do not need to adjust as they can be assigned with random values [9]. It can be trained in patch base or online by using OS-ELM. Fig. 3 shows the primal ELM network.

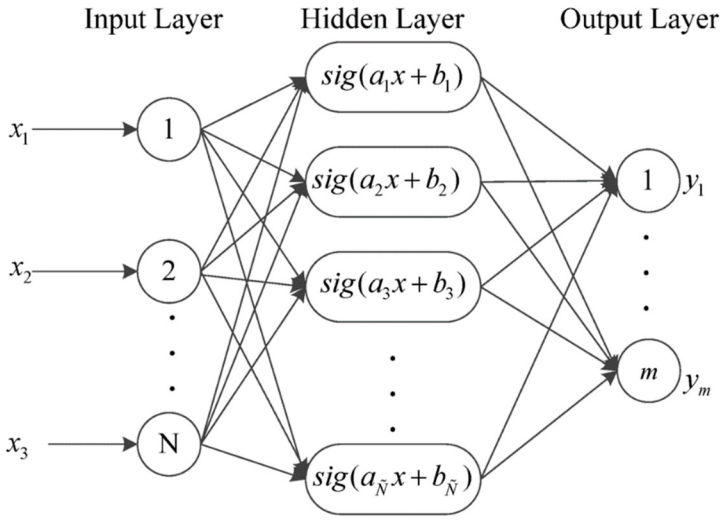

Fig. 3. Primal ELM Network

The output of ELM is:

$$f(x) = \sum_{i=1}^{L} \beta_i G(a_i, b_i, x) = \beta.h(x) \tag{3}$$

where $\beta_i =$ the output weight from the i-th hidden node, $G(a_i, b_i, x) =$ the output of the $i-th$ hidden node and $h(x) = [G(a_i, b_i, x),..., G(a_L, b_L, x)]^T$.

Given a training set $\{(x_i, t_i) \mid x_i \in R^d, t_i \in R^m, i = 1,..., N\}$, equation 3can be used to train the SLFN model

$$f(x_j) = \sum_{i=1}^{L} \beta_i G(\alpha_i, b_i, x_j) = t_j \tag{4}$$

where $j = 1,..., N$. Equation 4 can be written compactly as

$$H\beta = T \tag{5}$$

where

$$H = [h(x_i,..., h(x_N))]^T, h(x) = [G(\alpha_1, b_1, x),..., G(a_l, B_l, X)]^T, \beta = [\beta_1^T,..., \beta_L^T]^T \text{ and } T = [T_1^T,..., T_N^T]^T$$

then β can be estimated as $\beta = H^t T$. However, in this study, Online-Sequential Extreme Learning Machine (OS-ELM) is used for online learning purpose.

3.3. Online-sequential extreme learning machine

OS-ELM is a learning algorithm for feed forward networks with the ability to learn data one-by-one or chunk-by-chunk with fixed or varying block size [10]. In OS-ELM, the parameters of hidden nodes are randomly selected which is similar to ELM and the output weights of OS-ELM are analytically determined based on data that arrived in sequence. The algorithm of OS-ELM operates

in two phases which are the initialization phase and the sequential learning phase [10]. In initialization phase, a small block of training data is used to initialize the learning process:

$$\mathrm{N}_0 = \{(x_i, t_i)\}_{i=1}^{N_0} \tag{6}$$

where $\mathrm{N}_0 =$ given from a training set N. The initial estimate of the output weight, β is then calculated as

$$\beta^{(0)} = P_0 H_0^T T_0 \tag{7}$$

where

$$P_0 = \left(H_0^T H_0\right)^{-1}, T_0 = \left[t_1, \ldots, t_{N_0}\right]^T$$

and k is set to 0.

In the sequential learning phase, in order to calculate the partial hidden layer output matrix H_{k+1}, a new observation block of data at $k+1$ step is used. The output weight $\beta^{(k+1)}$ is then calculated as

$$\beta^{k+1} = \beta^{(k)} + P_{k+1} H_{K+1}^T \left(T_{k+1} - H_{k+1}\beta^k\right) \tag{8}$$

where

$$P_{k+1} = P_k - P_k H_{k+1}^T \left(I + T_{k+1} P_k H_{k+1}^T\right)^{-1} H_{k+1} P_k$$

k is then increase by 1 and the process is repeated for every new observation.

4. Methodology

Figure 4 shows the block diagram of the cascaded detector in TLD framework while Fig. 5 shows the block diagram of the cascaded detector in ELM-based detector with Haar-like feature.

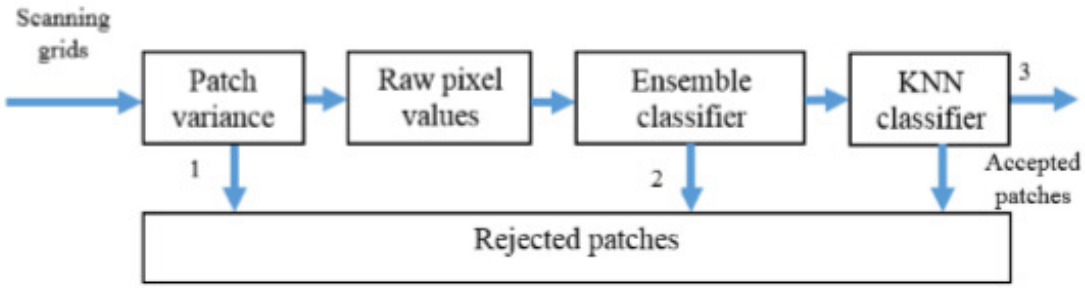

Fig. 4. Block Diagram of the Cascaded Detector in TLD Framework [6]

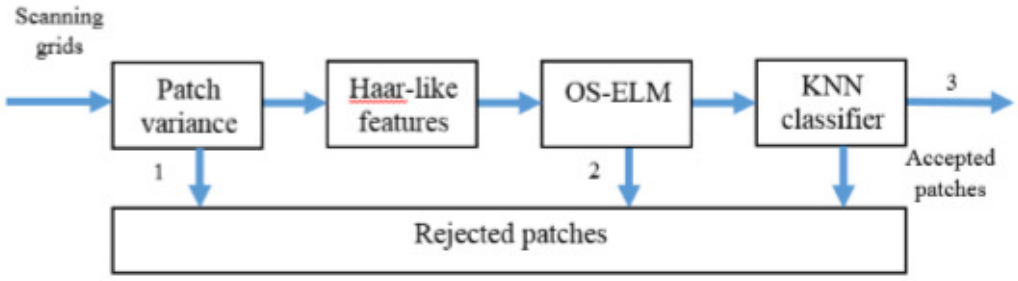

Fig. 5. Block Diagram of the Cascaded Detector in ELM-based Detector with Haar-like Feature

In our approach as shown in Fig. 5, the first stage which is the patch variance is the same as in the existing TLD where patches with a variance greater than 50% of the variance of the selected patch will be passed to the next stage while the rest will be rejected. Scanning window with different scales and shifts is used to generate the patches. The difference is that the features of the accepted patches are then extracted using a set of Haar-like features with different orientation as shown in Fig. 6.

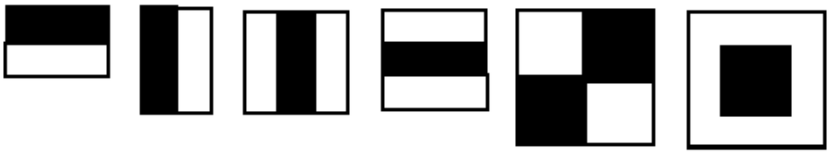

Fig. 6. Implemented Haar-Like Feature

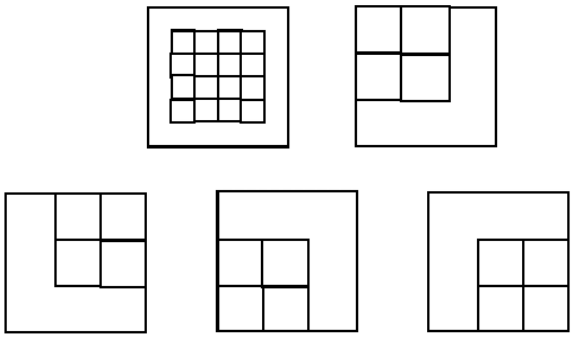

Fig. 7. Four Set of Rectangles

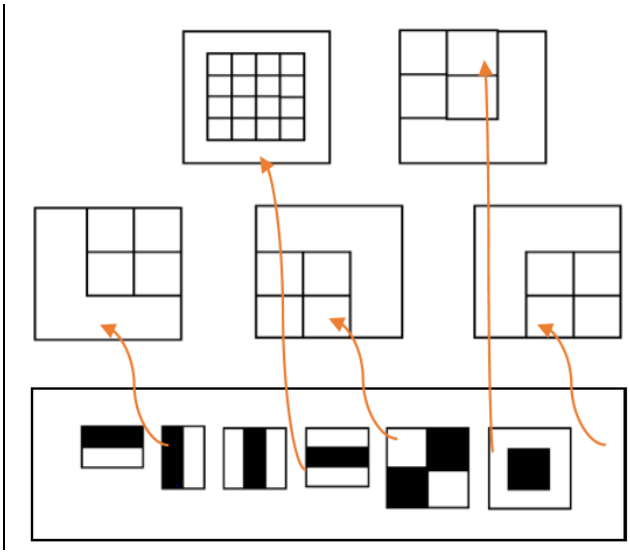

Fig. 8. Haar-like Feature

For each accepted patch, we used four set of rectangles to calculate Haar-like features as shown in Fig. 7. In each rectangle, we calculate the features by implement all the Haar-like features orientation shown in Fig. 6. Therefore, each accepted patch will have 192 features, as the total number of rectangles is 32 rectangles times with 6 Haar-like features orientations. This is illustrated in Fig. 8. These features are passed into the second stage which is the OS-ELM for training. As a result, from this stage, it gives several bounding boxes with possible representation of the object. In the third stage, K-NN is used to filter these bounding boxes based on some confidence criteria [11].

Basically, the third stage is responsible to classify the filtered patches or bounding boxes as object or non-object.

In general, the methodology of the proposed technique is shown in Fig. 7. The first step is to obtain the first frame image of the video. The second step is to initialize the structure of the Lucas-Kanade tracker, Haar-like feature and ELM-based detector such as number of hidden nodes and activation function. During the initialization of the tracker, a bounding box denotes an object and its movement between consecutive frames will be predicted. In detector, Haar-like feature extraction is implemented before train the ELM-based classifier. The third step is to obtain the next frame image and tracking and detection process of the target which have been initialized in the second step occur. In the fourth step, P-N expert will identify the negative and positive training set and tracking result which is the bounding box will be obtained. The process continues from the third step until the video ends or until the last frames.

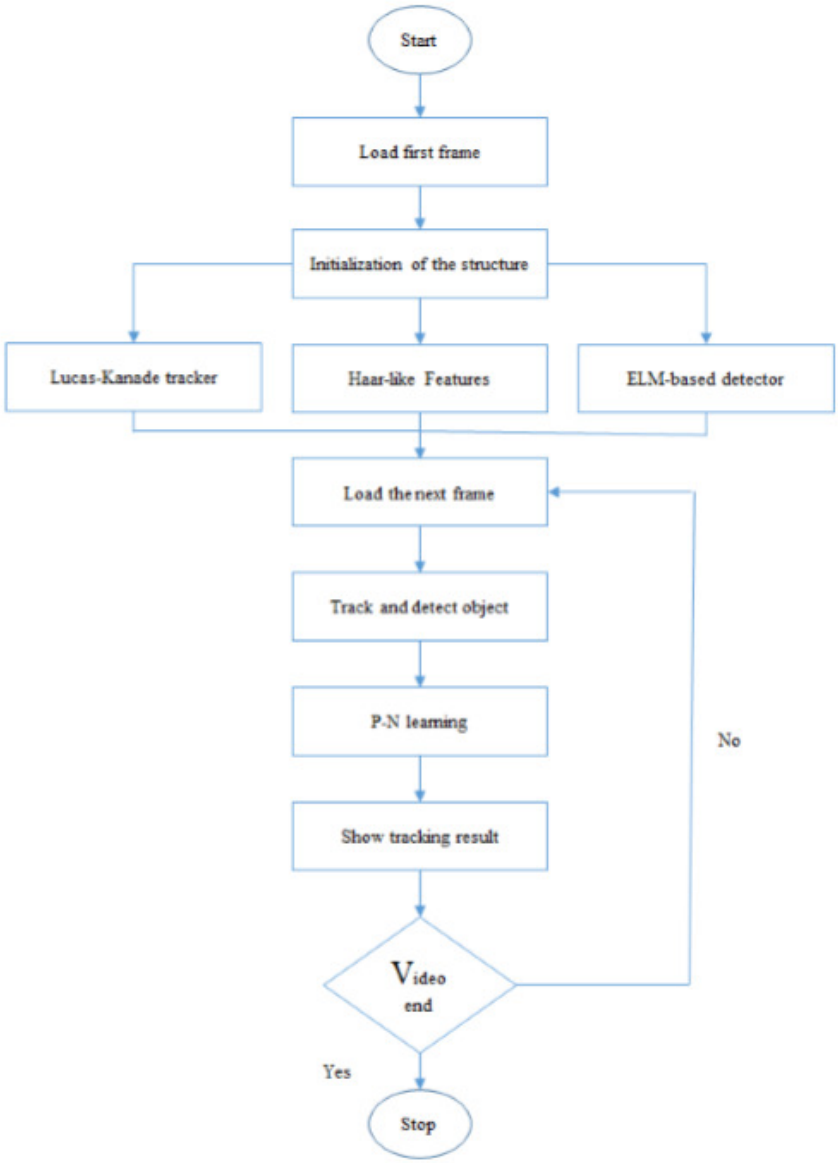

Fig. 9. Flowchart of the Methodology

Figure 9 shows the flowchart of the methodology that has been used throughout the experiment.

5. Experiments, results and discussion

Six videos from the TLD dataset as shown in Table 1 are used throughout the experiments. Besides that, the parameter setup for OS-ELM initialization is constant throughout the experiments and it is given in Table 2.

Table 1
TLD Dataset [11]

Name	**Properties of the Sequences**							
	Frame	Mov. Cam.	Part. Occ	Full occ.	Pose ch.	Illum ch.	Scale ch.	Similar objects
David	761	yes	yes	no	yes	yes	yes	no
Jump	313	yes	no	no	no	no	no	no
Ped.1	140	yes	no	no	no	no	no	no
Ped.2	338	yes	yes	yes	no	no	no	yes
Ped.3	184	yes	yes	yes	no	no	no	yes
Car	945	yes	yes	yes	no	no	no	yes

Table 2
Parameter Setup for OS-ELM

Parameter	**OS-ELM**
Variance threshold	50% of the variance of the chosen patch
K-NN threshold	0.65
K-NN validation	0.7
No. of hidden nodes	200, 500, 850
ELM threshold	0.5
Activation function	Sigmoid

In this experiments, all the parameters used are the same as in the released version [11]. Haar-like features are used for the patches instead of using the raw pixel values for the features. Besides that, as we are using OS-ELM to update the detector, choosing of training data and updating of the classifier are conducted in the same procedure as in TLD. In order to initialize the targeted object, the value of the bounding for each videos are used as it is given in the dataset. In order to evaluate the performance of the proposed approach, we used Precision, Recall and F-measure to compare the performance between our approach and the TLD framework with ensemble classifier.

Table 3
Performance Evaluation on TLD Dataset Measured by the Precision/ Recall/ F-measure

Sequence Video	**Tracker Name**			
	TLD	ELM200	ELM500	ELM850
David	0.94/0.86/0.89	**1.00/1.00/1.00**	**1.00/1.00/1.00**	1.00/1.00/1.00
Jump	1.00/0.80/0.88	0.72/0.72/0.72	0.72/0.72/0.72	0.72/0.72/0.72
Ped. 1	1.00/0.62/0.77	**1.00/1.00/1.00**	**1.00/1.00/1.00**	**1.00/1.00/1.00**
Ped. 2	0.75/0.55/0.63	0.89/0.70/0.78	**0.76/0.97/0.85**	0.56/0.71/0.63
Ped. 3	0.92/0.97/0.94	0.98/0.61/0.75	0.89/1.00/0.94	0.86/1.00/0.93
Car	0.97/0.97/0.97	0.98/0.61/0.75	0.94/0.83/0.88	0.94/0.83/0.88

Table 3 shows the performance evaluation on TLD dataset measured by Precision/ Recall /F-measure. The dataset was tested on three different hidden nodes. In general, Precision is the number of positive predictions divided by total number of positive class values predicted, Recall is the number of positive predictions divided by the number of positive class values in the test data while F-measure

conveys the balance between the Precision and Recall. The bold number indicate the best score of F-measure. The best video compared to TLD was Pedestrian 1 with F-measure of 1.00 compared to 0.77 in TLD. Besides that, the proposed approach offers better performance in terms of Precision in David, Pedestrian 1, Pedestrian 2 and Pedestrian 3. This approach offers better performance because of the good generalization property provided by ELM-based classifier. In addition, the ability of the OS-ELM is to update its output weights when new data arrives which enable it to adapt to any appearance changes in the object. Haar-like features also contribute to this better performance where measurements are taken at different scales using integral images instead of using raw pixel values and also this features representation is more robust against noise and light variations. Based on Table 3, Jumping shows the worst performance as compared to TLD. This may be due to the size of the initial bounding box which is smaller if compared to the other video. The fast movement of the object in the video also leads to this worst performance.

5. Conclusion

In this approach, ELM-based classifier with Haar-like features in TLD framework have been introduced to tackle the problem of tracking moving object under different conditions. OS-ELM is used to replace the ensemble classifier in the TLD framework because the ensemble classifier relies on simple pixel comparison hence, it offers weak generalization of the object. Based on the performance evaluation which was measured using Precision/ Recall/ F-measure, the best video in terms of F-measure is Pedestrian 1 with 1.00 as compared to 0.77 in TLD. Besides that, the proposed approach managed to give better performance in terms of Precision in David, Pedestrian 1, Pedestrian 2 and Pedestrian 3. These result was contributed by the good generalization property of the ELM-based detector and Haar-like features representation. However, this approach unable to offer better performance in some dataset. This may be caused by the condition of the video itself or large variance of the object appearance. In future, this approach can be improved by conducting the experiment with more number of hidden nodes instead of using three different number of hidden neurons as in our approach and further investigation on optimizing the number of hidden nodes should be done. In order to improve the frame processing rate, finding on better features extraction method may be done to represent the tracked object.

Acknowledgment

The author would like to thank Dr. Parvaneh Shahbanzade post-doctoral at Malaysia Japan International Institute of Technology for her contribution on guiding, correcting and monitoring the progress of writing this paper. Finally, thanks to Universiti Teknologi Malaysia and JICA for funding this research project through a Research Grant (R.K130000.7343.4B188) titled “Motor Imagery of Brain Computer Interface with Improved Common Spatial Pattern in Analyzing EEG Signal for Stroke Patients” and also Centre for Artificial Intelligent and Robotics ((CAIRO) - (U.K091303.0100.00000) for funding this research paper.

References

[1] Wu, Yi, Jongwoo Lim, and Ming-Hsuan Yang. "Online Object Tracking: A Benchmark." Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition, 2013.

[2] Jepson, Allan D, David J Fleet, and Thomas F El-Maraghi. "Robust Online Appearance Models for Visual Tracking." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25, no. 10 (2003): 1296-1311.

[3] Babenko, Boris, Ming-Hsuan Yang, and Serge Belongie. "Visual Tracking with Online Multiple Instance Learning." Paper presented at the Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, 2009.

[4] Lasserre, Julia A, Christopher M Bishop, and Thomas P Minka. "Principled Hybrids of Generative and Discriminative Models." Paper presented at the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), 2006.

[5] Grabner, Helmut, and Horst Bischof. "On-Line Boosting and Vision." Paper presented at the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), 2006.

[6] Maliha, Ayman, Rubiyah Yusof, and Ahmed Madani. "Online Sequential-Extreme Learning Machine Based Detector on Training-Learning-Detection Framework." Paper presented at the Control Conference (ASCC), 2015 10th Asian, 2015.

[7] Jia, Chunxiao, Zhongli Wang, Xian Wu, Baigen Cai, Zhenhui Huang, Guiguo Wang, Tianbai Zhang, and Dezhong Tong. "A Tracking-Learning-Detection (Tld) Method with Local Binary Pattern Improved." Paper presented at the 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2015.

[8] Piao, Songlin, and Karsten Berns. "Multi-Object Tracking Based on Tracking-Learning-Detection Framework." *Field and Assistive Robotics—Advances in Systems and Algorithms* (2014): 74-87.

[9] Hoang, Minh-Tuan T, Hieu T Huynh, Nguyen H Vo, and Yonggwan Won. "A Robust Online Sequential Extreme Learning Machine." Paper presented at the International Symposium on Neural Networks, 2007.

[10] Liang, Nan-Ying, Guang-Bin Huang, Paramasivan Saratchandran, and Narasimhan Sundararajan. "A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks." *IEEE Transactions on Neural networks* 17, no. 6 (2006): 1411-1423.

[11] Kalal, Zdenek, Krystian Mikolajczyk, and Jiri Matas. "Tracking-Learning-Detection." *IEEE transactions on pattern analysis and machine intelligence* 34, no. 7 (2012): 1409-1422.