Chapter

Deep Siamese Networks toward Robust Visual Tracking

Mustansar Fiaz, Arif Mahmood and Soon Ki Jung

Abstract

Recently, Siamese neural networks have been widely used in visual object tracking to leverage the template matching mechanism. Siamese network architecture contains two parallel streams to estimate the similarity between two inputs and has the ability to learn their discriminative features. Various deep Siamese-based tracking frameworks have been proposed to estimate the similarity between the target and the search region. In this chapter, we categorize deep Siamese networks into three categories by the position of the merging layers as late merge, intermediate merge and early merge architectures. In the late merge architecture, inputs are processed as two separate streams and merged at the end of the network, while in the intermediate merge architecture, inputs are initially processed separately and merged intermediate well before the final layer. Whereas in the early merge architecture, inputs are combined at the start of the network and a unified data stream is processed by a single convolutional neural network. We evaluate the performance of deep Siamese trackers based on the merge architectures and their output such as similarity score, response map, and bounding box in various tracking challenges. This chapter will give an overview of the recent development in deep Siamese trackers and provide insights for the new developments in the tracking field.

Keywords: Siamese networks, visual object tracking, deep learning, neural network, end-to-end learning

1. Introduction

In the past few decades, visual object tracking (VOT) has become a promising and attractive research field in computer vision area. It became popular among researchers due to its wide range of applications including autonomous vehicles [1, 2], surveillance and security [3, 4], traffic flow monitoring [5, 6], human computer interaction [7, 8] and many more. Popularity in the field is because of various tracking challenges and opportunities. In recent years, researchers have made remarkable endeavors and developed a number of state-of-the-art trackers to handle various tracking challenges. Despite the fact that significant progress has been made in the field but still trackers have not achieved consummate performance and VOT is still an open challenge yet to be fully addressed. Various challenges to be handled by VOT include fast motion, motion blur, occlusion, deformation, illumination variations, background clutter, in- or out-planer rotations, out-of-view, low resolution, and scale variations.

The objective of VOT is to identify a region of interest in video frames. VOT consists of four sequential components such as target initialization, target appearance modeling, motion estimation, and target localization. In target initialization, the region of interest is annotated using any of the representations including ellipse, centroid, object silhouette, object skeleton, object contour, or object bounding box. In generic object tracking, the position of the region of interest as the target is given in the first frame of a video and the tracking algorithm predicts the target location in the rest of the frames. The target appearance model represents a better target feature representation and a mathematical model to identify the region of interest using learning methodologies. While the target motion estimation module predicts the position of the target in sequential frames by either greedy search or maximum posterior prediction. The tracking problem is simplified as the constraints applied over the target appearance model and motion estimation. During tracking, both appearance and motion models are updated to capture the new target appearance and its behavior.

In this chapter, we focus on monocular, casual, model-free, short-term, and single-target trackers. The causality means that a tracker has the ability to estimate the target location in the current frame without prior information of the future frames. While model-free characteristic stands for supervised learning where target bounding box is given in the first frame of the video. Finally, short-term denotes that during tracking, a tracker is unable to re-detect the target once it is lost.

The performance of the trackers is highly affected by the feature representations. Features are broadly classified into hand-crafted (HC) and deep features. Traditional features are known as HC features such as histogram of oriented gradients (HOG), local binary patterns (LBP), color names and scale-invariant feature transform, etc. Nowadays, computer vision researchers are selecting deep features for better representation. Deep features are more capable to capture multi-level information and to encode the target appearance variant features compared to HC features. Deep features are extracted using different methods such as convolutional neural networks (CNN) [9], recurrent neural networks (RNN) [10], auto-encoder [11], residual networks [12], and generative adversarial networks (GAN) [13] for different computer vision applications.

In recent years, CNN-based methods have been adopted in various computer vision tasks and gained popularity due to improved performance in face verification [14], image classification [15], semantic segmentation [16], medical image segmentation [17], object detection [18], etc. An empirical and comprehensive study performed by Fiaz et al. [19] showed that deep trackers have shown an improved performance compared to HC feature-based trackers. The discriminative power of state-of-the-art deep trackers is explored by employing deep features. It is difficult to train a discriminative deep tracker efficiently due to data-hungry property. Various deep trackers are developed to handle scarce training data problem by employing shallow features extracted from pre-trained off-the-shelf models such as AlexNet [20], VGGNet [9], etc. Nevertheless, these approaches do not fully benefit from end-to-end learning. Deep trackers that apply stochastic gradient descent (SGD) methods are not real-time because they take a lot of time to fine-tune the multiple layers of the network.

In order to handle those restrictions, a simple advocate approach known as Siamese network is utilized to compute the similarity between the two input images. Siamese networks are trained offline to learn the similarity between two input images and are evaluated online without fine-tuning for new target estimation. In this chapter, we study different types of Siamese networks developed for tracking. We also present an experimental study to analyze the performance of the Siamese trackers over OTB2013 [21] and OTB2015 [22] benchmarks.

2. Related work

In the literature, there exist many comprehensive studies on VOT. Each study focuses on specific research aspects going on in the field. Fiaz et al. [19] classified the tracking algorithms into correlation and noncorrelation filter-based trackers. An extensive experimental study was performed over hand-crafted and deep feature trackers. Similarly, Li et al. [23] also studied the deep trackers and categorized deep trackers into three classes including network structure, network function, and network training. Leang et al. [24] discussed single target trackers while Zhang et al. [25] performed their study over the sparse trackers. Yang et al. [26] focused on the context information by considering auxiliary objects as the target context of the tracking object.

These studies have been performed by tireless efforts made by the research community and developed various state-of-the-art trackers. The tracking algorithms can be classified as tracking by detection, discriminative correlation filters, deep convolutional neural networks, and Siamese network-based trackers.

2.1 Tracking by detection-based trackers

In many tracking algorithms, classifiers are considered as the fundamental part to discriminate the target object from nontarget objects such as support vector machine (SVM), random decision forest, as well as various boosting-based classifiers. Classifiers are updated to integrate the new target appearance during online learning in various tracking by detection algorithms. For example, multiple instance learning framework proposed by Babenko et al. [27] employed gradient boosting to learn the classifiers. Hare et al. [28] utilized structured output to estimate the target location and employed SVM for online adaptive tracking. Zhang et al. [29] applied Bayes classifiers for online adaptation of the target over a multi-scale feature space built on a data-dependent basis.

2.2 Discriminative correlation filter-based trackers

The development of trackers based on correlation filters has boosted the tracking performance. Bolme et al. [30] proposed a fast tracker by minimizing the sum of squared error (SSE) between the actual output and the desired output in the frequency domain. Kernelized correlation filters (KCF) [31] utilized the multi-channel features using circulant matrices in the Fourier domain and used the Gaussian kernel function to discriminate a target from the background. The discriminative correlation filter trackers have their own limitations such as they require to fix model and patch sizes. A model may learn undesired information resulting in reduced performance. SRDCF [32] introduces a spatial regularization method in discriminative correlation trackers to reduce the effect of background information by penalizing it. SRDCFdecon proposed by Danelljan et al. [33] tackled the contaminated training samples to improve robustness. Li et al. [34] proposed STRCF that integrates the temporal regularization in SRDCF using a passive-aggressive algorithm to improve the tracking performance. CSRDCF [35] incorporates the channel and spatial reliability within correlation filters. CSRDCF integrates the spatial reliability using a spatial binary map at the target location, while the channel reliability by estimating the channel and detection reliability metrics.

2.3 Deep convolutional neural network-based trackers

Deep convolutional neural networks have presented an outstanding performance in many computer vision applications. Deep learning has limitations due to limited training data and high computational cost. However, much progress has been made and many state-of-the-art deep trackers have been proposed. Nam and Han employed CNN to develop a multi-domain adaptive deep tracker [36]. Nam et al. [37] integrated CNN in a tree structure to model the target appearance. A tree is constructed from multiple hierarchical CNN-based target appearances. Ma et al. [38] exploited the rich hierarchical deep features using correlation filters. Qi et al. [39] hedged the weak classifiers and obtained a strong classifier by captivating the benefit from multi-level deep features.

2.4 Template matching-based trackers

Tracking by matching is one of the most basic concepts in tracking where target pixels are directly compared with the input patches from the video. Briechle and Hanebeck [40] introduced the simplest template matching mechanism in tracking via a normalized cross-correlation. TLD-tracker [41] also employs normalized cross-correlation mechanism. Later on, many template matching trackers focused on distorted tracking objects. Wang et al. [42] performed matching using superpixels. Nguyen and Smeulders [43] used color invariants to discriminate targets from the background. Godec et al. [44] employed HOG features for probabilistic matching. Held et al. [45] used deep regression networks for matching. Bertinetto et al. [46] exploited fully convolutional features to compute the correlation between the target and the search patches.

In this section, we noticed that various tracking algorithms have been proposed to solve the tracking problem but still research area is active. We also observed that there exist different comprehensive surveys that focus on various tracking frameworks. On the contrary, we present a study on Siamese networks employed in tracking. We categorized the Siamese trackers into three categories. Moreover, we also evaluated the robustness of the different Siamese trackers.

3. Siamese networks for tracking

In correlation filter-based trackers, a response map is computed between a target template and a candidate patch in the Fourier domain. In object tracking, the center of the target is focused and a weight matrix w is trained such that it minimizes the squared error from the target y. The tracking problem can be defined as a regression problem which depicts a closed-form solution and is formulated as

$$\|Bw - y\|_2^2 + \lambda \|w\|_2^2, \tag{1}$$

where B is the search space feature vectors, λ is a regularization parameter, and || . ||2 means the ℓ_2-norm of a vector. The solution for Eq. (1) is described as:

$$w = \left(B^T B + \lambda I\right)^{-1} B^T y. \tag{2}$$

Since Eq. (2) has high computational cost due to inverse matrix computation, thus cannot be used directly for tracking. Hence, the described problem can be resolved in the dual form as follows:

$$w = B^T \alpha, \tag{3}$$

where α denotes the discriminatory part. For tracking problems, the challenge is to optimize α in dual form solution in Eq. (3).

Another alternative approach is to learn a similarity function to compare the similarity between the template image and the candidate image. A Siamese network

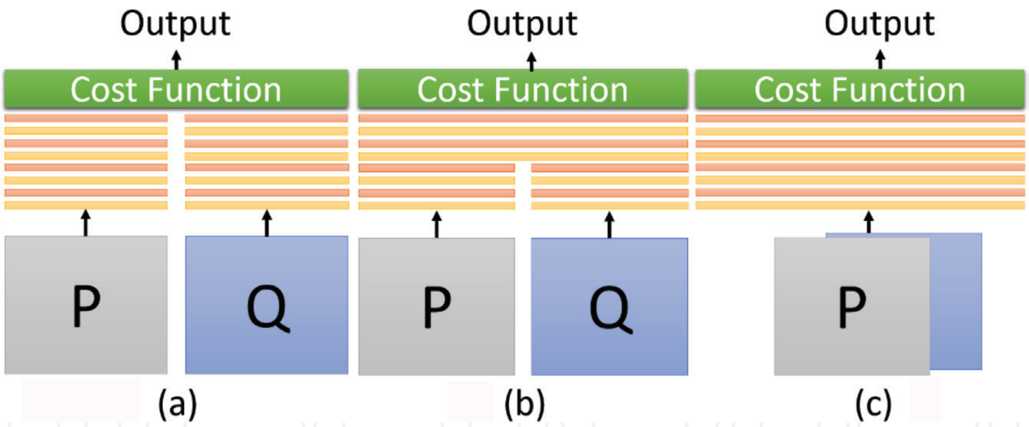

Figure 1.
Types of Siamese networks (a) Late merge, (b) Intermediate merge and (c) Early merge.

architecture is a Y-shaped network that takes two images as inputs and returns similarity as output. Siamese networks determine if the two input images have identical patterns or not. The concept of Siamese was initially introduced for signature verification and fingerprint recognition, and later adapted in many computer vision applications such as large scale video classification [47], stereo matching [48], face recognition and verification [49], and patch matching [50] etc. A series of state-of-the-art Siamese-based trackers have been proposed in the past few years. We observe that Siamese-based trackers utilize embedded features by employing CNN to compute the similarity. By analyzing the architecture of deep Siamese trackers, we classify them into three categories based on layer position of the merge; (i) late merge, (ii) intermediate merge, and (iii) early merge architectures as shown in **Figure 1**.

- Late merge: the input images are processed separately by two individual parallel networks and are merged at the last layer of the network (**Figure 1(a)**).

- Intermediate merge: the input images are processed separately in the initial part of the network and then merged well before the final layer (**Figure 1(b)**).

- Early merge: the input images are stacked before feeding to the network and then a unified input is fed forward to the network for inference (**Figure 1(c)**).

We also observe that Siamese-based trackers produce different types of output such as similarity score, response map, and bounding box. Siamese-based trackers with similarity score as output mean that they return the similarity as probability measure, whereas the response map means a two-dimensional similarity score map. The maximum value in the similarity map represents the location of maximum similarity between two patches and low value for the dissimilar region. Some Siamese-based trackers directly yield the bounding box location of the target.

3.1 Siamese late merge trackers

This subsection studies the tracker where the two input images are fed forward to two separate CNN models and are merged at the final layer to get the final response.

3.1.1 SINT

Siamese instance search tracker (SINT) is proposed by Tao et al. [51]. SINT learns an offline matching function and estimates the best-matched patch for incoming

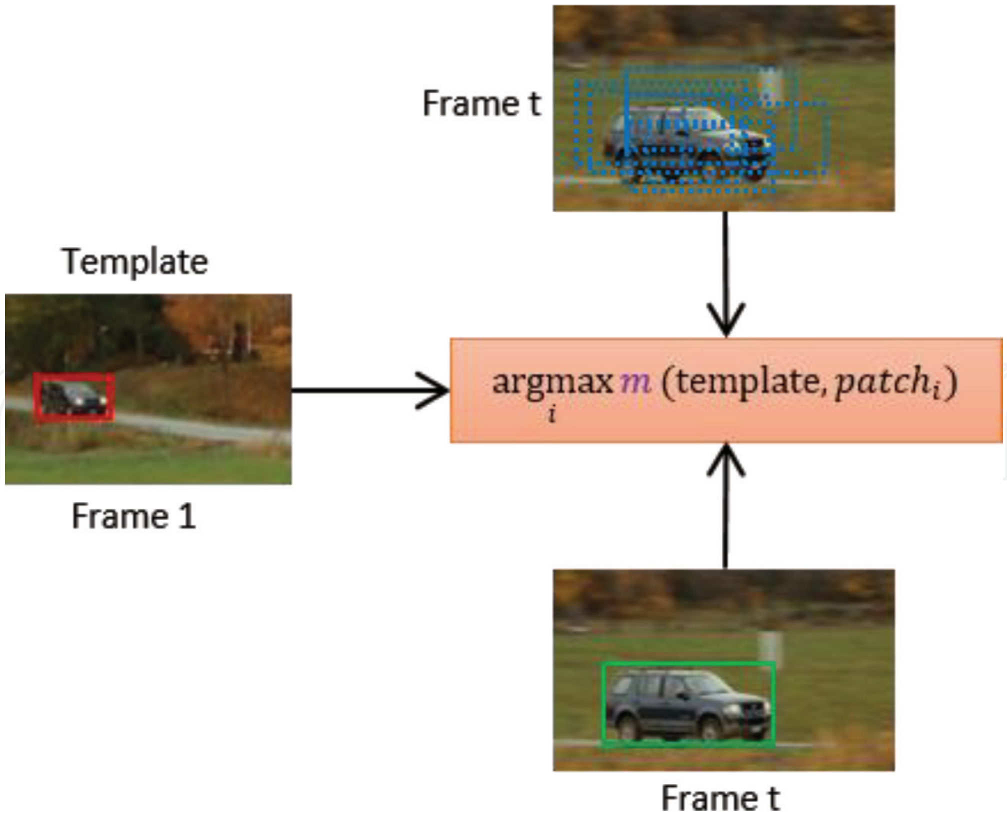

Figure 2.
SINT tracking framework [51].

frames in a video (**Figure 2**). The architecture of SINT consists of two streams including query stream and search stream. Each stream is composed of 5 convolutional layers, 3 region-of-interest pooling layers, and 1 fully connected layer. Both query and search streams are merged using a matching function known as contrastive loss function. The matching function is responsible to differentiate the background information from the target. The SINT is trained offline by giving template patch at query branch and candidate patches at the stream branch. During tracking, SINT does not update its weight parameters and template patch at query branch is matched with the candidate patches at the stream branch for each incoming frame. The SINT estimates the best-matched patch based on maximum score. A ridge-bounding box regression is employed to refine the bounding box.

3.1.2 SiameseFC

Siamese fully convolutional network (SiameseFC) proposed by Bertinetto et al. [46] addresses the general similarity learning between the target image and search image as shown in **Figure 3**. During training, SiameseFC exploits the deep features using embedding functions and learns the similarity between the two images. During tracking, SiameseFC takes two images and infers a response map. The new target position is estimated at the maximum value on the response map where input images have the maximum similarity.

3.1.3 CFNet

Valmadre et al. [52] proposed correlation filter network (CFNet) by adding two layers including correlation filter and crop layer within SiameseFC template branch which makes it more shallower but efficient. While SiameseFC learns the unconstrained features to estimate the similarity score, CFNet learns the discriminative features

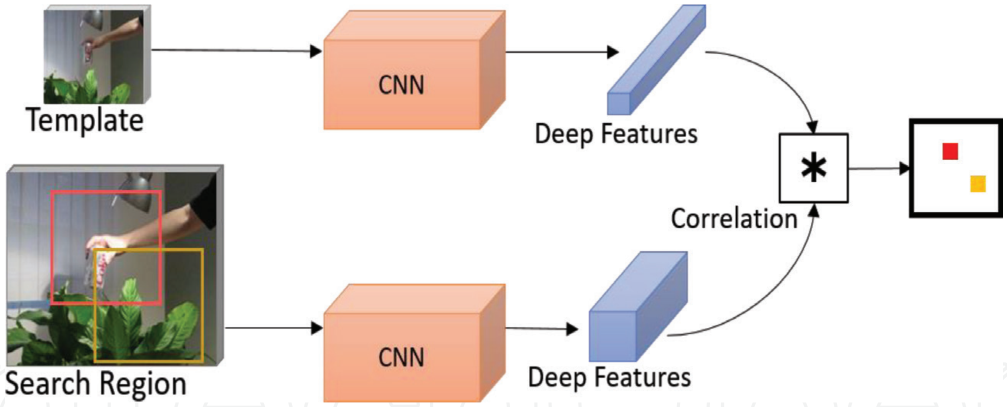

Figure 3.
SiameseFC architecture [46].

using correlation filter layer and solves the ridge regression problem via exploiting the negative samples in the search region. Similar to SiameseFC, CFNet is trained offline and weight parameters are fixed during tracking. CFNet produces a response map for template and search region with a high value representing the maximum similarity.

3.1.4 SIAMRPN

Li et al. [53] proposed a Siamese region proposal network (SIAMRPN) in order to improve the robustness compared to SiameseFC and CFNet. Both SiameseFC and CFNet do not employ bounding box regression and thus require multi-scale testing. SIAMRPN integrates region proposal network (RPN) within SiameseFC which makes it more elegant. The concept of RPN was introduced in Faster RCNN [18]. RPN has capability to extract more precise and efficient proposals due to the supervision of bounding box regression and binary classifier.

SIAMRPN consists of two components including Siamese network and RPN as shown in **Figure 4**. Siamese network is responsible for feature computation. Its template branch takes z as target patch and gives $\varphi(z)$ as output target features while detection branch requires x search image and returns $\varphi(x)$ as search region features. Whereas RPN is composed of a pairwise correlation module and a supervision module. The supervision module has two outputs consisting of a binary classifier and a bounding box regressor. If there are k anchors, the pairwise correlation module increases the channels for $\varphi(z)$ using convolution layers by 2k for classification denoted as $([\varphi(z)]_{cls})$ and 4k for regression represented as $([\varphi(z)]_{reg})$. The search region features $\varphi(x)$ are also divided into $[\varphi(x)]_{cls}$ and $[\varphi(x)]_{reg}$ branches using convolutional layers while the number of channels for $\varphi(x)$ is kept unchanged. A correlation operation is performed for both classification and regression branches by considering $\varphi(z)$ as correlation kernel in a group manner. It means that the channel number of a group $\varphi(z)$ is equal to the number of the channel $\varphi(x)$. The SIAMRPN is trained using Stochastic Gradient Descent (SGD) method to optimize the following loss function:

$$loss = L_{cls} + \lambda L_{reg}, \tag{4}$$

where L_{cls} represents the classification loss which is a cross entropy loss function and L_{reg} means bounding box regression loss, and λ is a balancing parameter.

3.2 Siamese intermediate merge trackers

This section describes the tracking models where the two input images are input separately to the network and are merged somewhere before the final layer of the CNN.

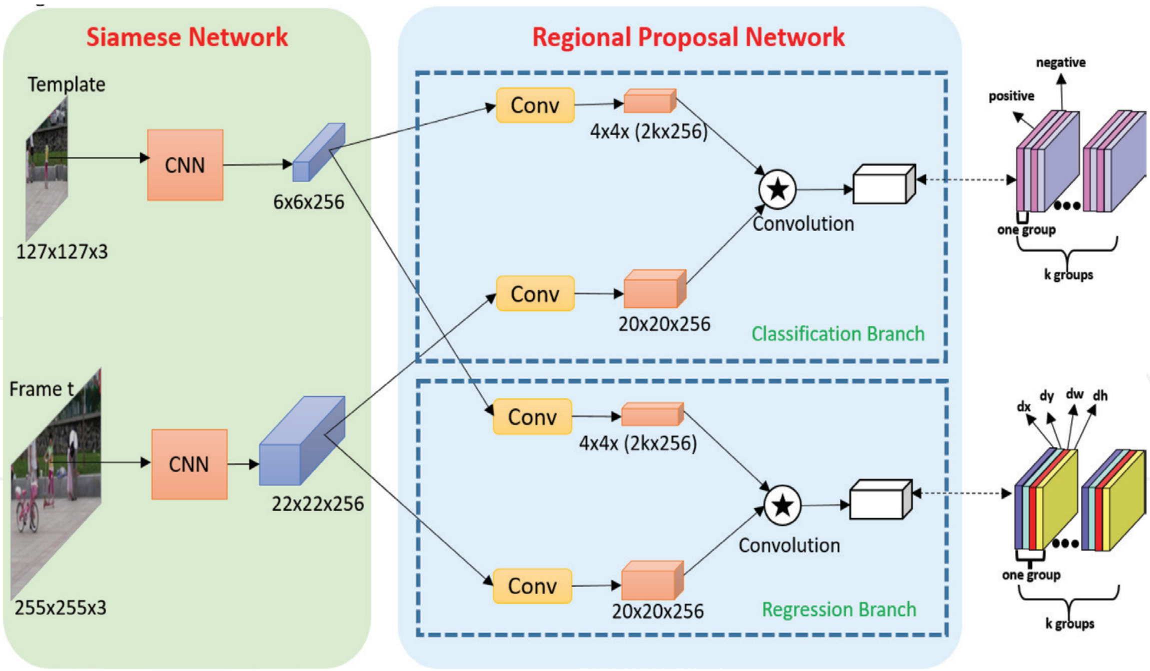

Figure 4.
SIAMRPN architecture [53].

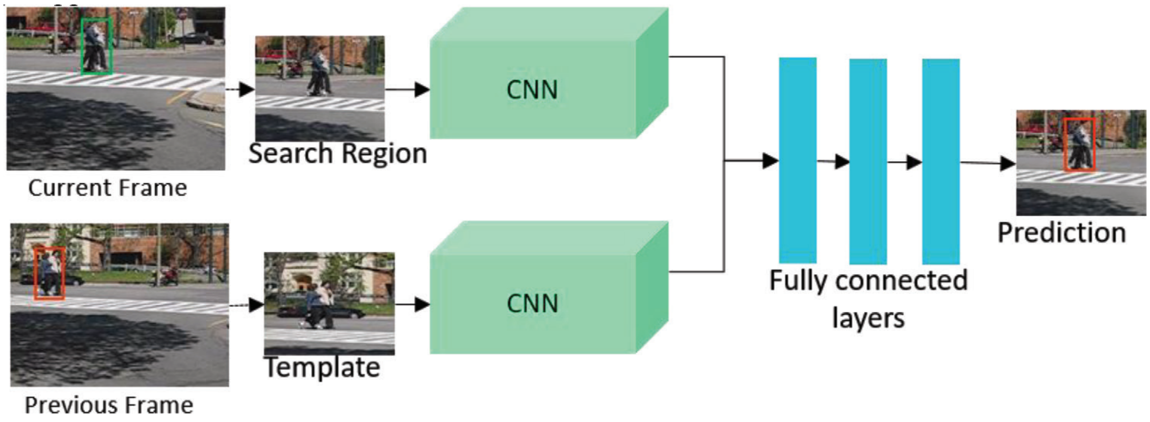

Figure 5.
GOTURN tracking framework [45].

3.2.1 GOTURN

Held et al. [45] proposed generic object tracking using regression network (GOTURN) and exploited the target appearance and motion relationships. GOTURN predicts the new target object for the current frame by taking the template image from the previous frame. Both input images are cropped with the background region for prediction as demonstrated in **Figure 5**. GOTRUN consists of two streams of 5 convolutional layers for both template and search images. The template and search streams are fused and feed-forwarded to three shared fully connected layers. During tracking, GOTURN directly regresses the target position and does not update the weight parameters to adapt the new target appearances.

3.2.2 YCNN

Chen and Tao [54] proposed the YCNN tracker to estimate the similarity between two input images. YCNN model consists of two separate 3 convolutional layers and two shared fully connected layers. The target object and search images are fed forward two separate 3 convolutional layers and then merged before forwarding to two shared fully connected layers. The output of YCNN is a response map. The network is trained end-to-end using Gaussian map as a label with the maximum value at the center. During tracking, the maximum position on the confidence map gives the new target position. The drift problem is handled by averaging the maximum five confidence values, while the scale problem is tackled by repeating the inference with different template sizes.

3.2.3 EAST

Huang et al. [55] proposed early stopping tracker (EAST) to exploit similarity between the two input images and learn the different policies by employing Reinforcement Learning (RL) to improve the accuracy while maintaining high speed. On the contrary to SiameseFC, EAST infers the new target position in single evaluation on original template size. The tracking problem is formulated as Markov decision process. The network agent is trained offline such that agent decides whether the target object has high confidence on early layers or continue to go deep by processing subsequent layers to obtain the maximum confidence for each frame. Agent makes a decision based on early stopping criterion for each layer.

3.3 Siamese early merge trackers

In this subsection, we study the tracking models where the input images are aggregated or stacked before feeding to the network.

3.3.1 CNNSI

Fiaz et al. [56] proposed CNN with structural input (CNNSI) to exploit the deep discriminative features to learn the similarity between the target and candidate patches as shown in **Figure 6**. The target and candidate images are stacked together and feed-forwarded to the network to get the similarity and dissimilarity scores. The CNNSI is trained offline end-to-end using SGD method to learn the similarity. During the tracking, target and candidate patches are stacked and fed to the network to get similarity and dissimilarity scores for all the candidate patches. The maximum similarity score yields the new target position. The bounding boxes are refined using

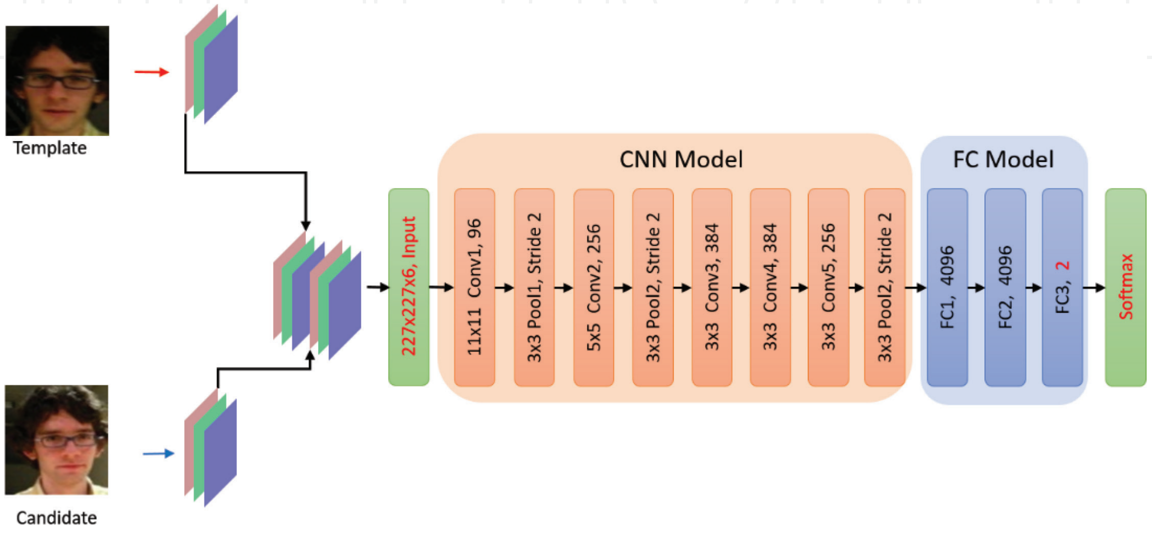

Figure 6.
CNNSI network architecture [56].

a bounding box regressor which is trained on the first frame of the sequence. Short-term and long-term updates are performed to integrate the new target appearance.

3.3.2 SiameseCNN

Taixé et al. [57] presented a Siamese CNN (SiameseCNN) for pedestrian tracking to exploit the pedestrian appearance and geometrical position. The proposed network requires a stack of two target images along with their optical flow and forwarded to three CNN layers and three fully connected layers. The network is trained using a gradient boosting classifier to predict the final trajectory of the pedestrian. For negative samples, contextual features along with relative geometry are provided to train the classifier. To infer the pedestrian, the gradient boosting classifier makes the final decision based on the maximum score.

4. Experimental analysis

This section discusses the experimental results and analysis over the OTB2013 [21] and OTB2015 [22] benchmarks. The OTB2013 consists of 50 different sequences having 11 challenges including fast motion (FM), background clutter (BC), motion blur (MB), low resolution (LR), scale variation (SV), in-plane rotation (IPR), out-plane rotation (OPR), deformation (DEF), occlusion (OCC), illumination variation (IV), and out-of-view (OV). OTB2015 contains 100 videos, which is an improved version of OTB2013 having all the challenges from OTB2013.

The Siamese trackers are evaluated using precision, success, and speed measures. One pass evaluation (OPE) is utilized to evaluate the robustness of the Siamese trackers. Performance of the trackers is illustrated using precision and success graphs. Euclidean distance is calculated between the ground-truth center and predicted centers to compute the precision as:

$$\varphi_{tp} = \sqrt{(x_t - x_p)^2 + (y_t - y_p)^2}, \tag{5}$$

where (x_t, y_t) and (x_p, y_p) shows the ground-truth center and predicted center in a frame respectively. A frame is measured as successful if the value of φ_{tp} is less than a threshold else not. The precision threshold value is set to 20 pixels. The target changes its size in a sequence and precision only considers the pixel difference of the center of the target. Thus precision does not a true picture of the target shape. Hence, a more robust success metric is employed for evaluation of trackers. An overlap score (OS) is calculated between the ground-truth and predicted bounding box to compute success as:

$$O_s = \frac{|b_t \cap b_p|}{|b_t \cup b_p|}, \tag{6}$$

where b_t represents the bounding box for ground-truth, b_p denotes the predicted bounding box, |.| shows the number of pixels, ∩ means intersection and ∪ shows the union operator. The O_s determines that a frame is successful or not. If O_s is less than a threshold then that frame is referred to as a successful frame and vice-versa. The overlap score for success varies between 0 and 1, and the threshold is set at 0.5. For precision and success, average precision and average success scores are reported by computing the mean of precision and OS for all the frames in a benchmark respectively. The speed of the Siamese trackers is reported in frames-per-second (FPS) by computing the mean of speed for all the frames in a benchmark.

For comparison of different Siamese architectures, we carefully selected Siamese trackers such that at least one tracker is selected from each category. The selected trackers are SINT [51], SiameseFC [46], CFNet [52], SIAMRPN [53], GOTURN [45], and CNNSI [56]. All results are reported from the original authors except, the GOTURN because the authors did not report results over the selected benchmarks.

4.1 Quantitative evaluation

In this subsection, we discuss the quantitative comparison of Siamese Trackers.

4.1.1 Overall performance

Figures 7 and 8 and **Table 1** show the precision and success of selected Siamese trackers over OTB2013 and OTB2015 respectively. The precision and success graphs show that SIAMRPN achieved outstanding performance compared to the other trackers. We also observe that the rank of the trackers does not change with respect to precision and success for both benchmarks. GOTURN does not perform well as compared to the other Siamese trackers.

4.1.2 Challenge-based evaluation

We also evaluated the performance of Siamese trackers for eleven different tracking challenges over OTB2015 benchmark. **Figures 9** and **10** and **Tables 2** and **3** show the performance of Siamese trackers using precision and success respectively. We observe that SIAMRPN attained better performance for all the tracking challenges

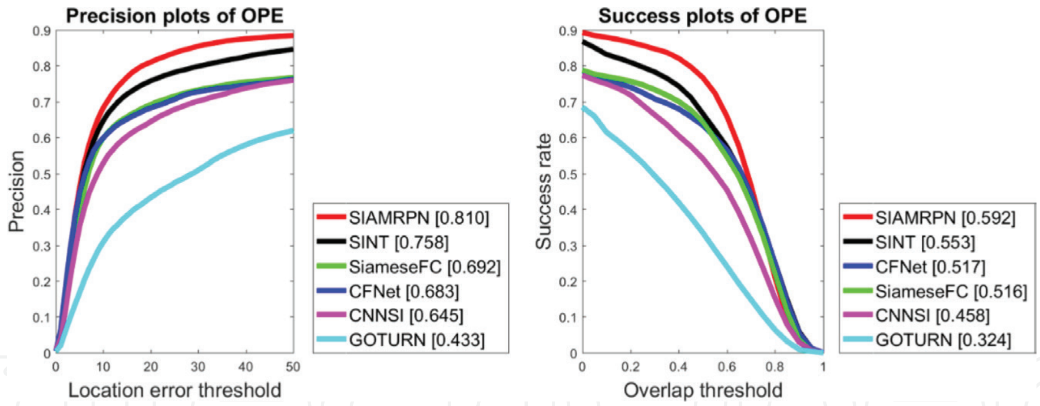

Figure 7.
Precision and success plots over OTB2013.

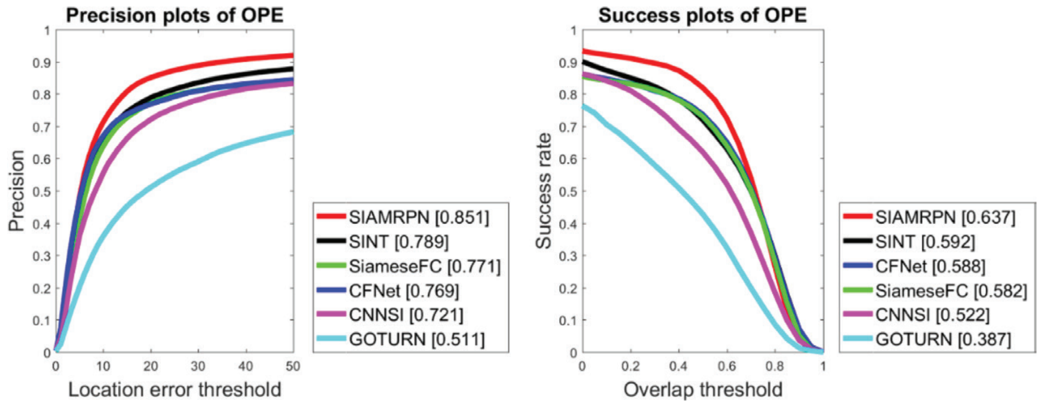

Figure 8.
Precision and success plots over OTB2015.

	Trackers	SIAMPRN	SINT	SiameseFC	CFNet	CNNSI	GOTURN
OTB2013	Precision	**81.0**	75.8	69.2	68.3	64.5	43.3
	Success	**59.2**	55.3	51.6	51.7	45.8	32.4
OTB2015	Precision	**85.1**	78.9	77.1	76.9	72.1	51.1
	Success	**63.7**	59.2	58.8	58.2	52.2	38.7
Speed (fps)		160	4	86	43	0.53	**165**

Table 1.
Comparison of Siamese trackers over OTB2013 and OTB2015 benchmarks.

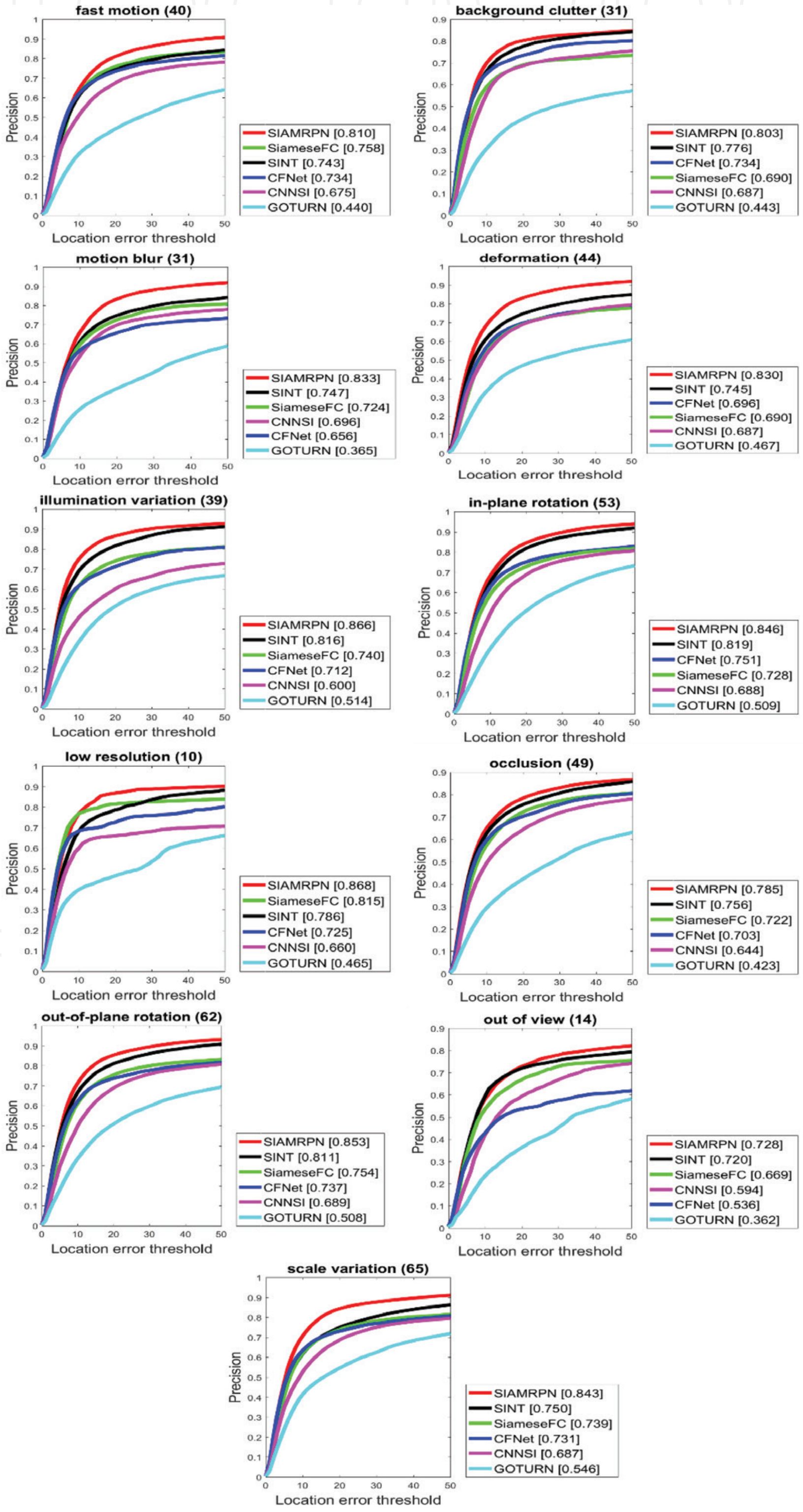

Figure 9.
Precision plots for eleven tracking challenges over OTB2015.

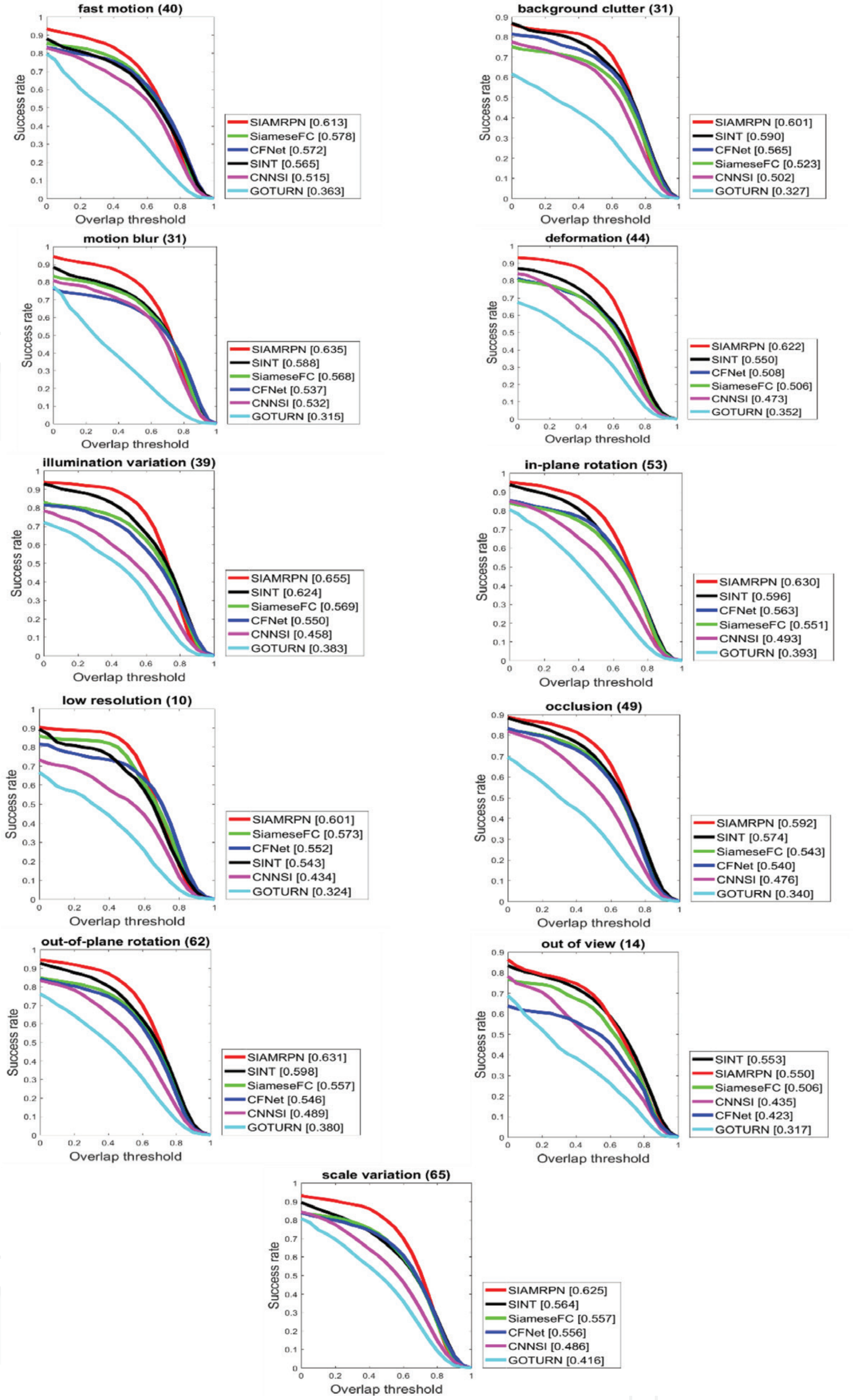

Figure 10.
Success plots for eleven tracking challenges over OTB2015.

using both precision and success. While GORTURN does not show good performance and ranked at the last. We noted that SiameseFC exhibited better performance after SIAMRPN for fast motion and low-resolution challenges while SINT ranked second best for the rest of the challenges handling those challenges more efficiently.

4.2 Qualitative evaluation

Qualitative study of Siamese-based trackers has performed over five different videos including *Bolt*, *ClifBar*, *FaceOcc1*, *Jogging-1*, and *CarScale* shown in **Figure 11**. The *Bolt* video depicts OCC, DEF, IPR and OPR challenges. Trackers such as SiameseFC, CFNet,

Trackers	SIAMRPN	SiameseFC	SINT	CFNet	CNNSI	GORTURN
FM	81.0	75.8	74.3	73.4	67.5	44.0
BC	80.3	69.0	77.6	73.4	68.7	44.3
MB	83.3	72.4	74.7	65.6	69.6	36.5
DEF	83.0	69.0	74.5	69.6	68.7	45.7
IV	86.8	74.0	81.6	71.2	60.0	51.4
IPR	84.6	72.8	81.9	75.1	68.8	50.9
LR	86.8	81.5	78.6	72.5	66.0	45.6
OCC	78.5	72.2	75.6	70.3	64.4	42.3
OPR	85.3	75.4	81.1	73.7	68.9	50.8
OV	72.8	66.9	72.0	53.6	59.4	36.2
SV	84.3	73.9	75.0	73.1	68.7	54.6

Table 2.
Precision of Siamese tracker over different challenges.

Trackers	SIAMRPN	SiameseFC	SINT	CFNet	CNNSI	GORTURN
FM	61.3	57.8	56.5	57.2	51.5	36.3
BC	60.1	52.3	59.0	56.5	50.2	32.7
MB	63.5	56.8	58.8	53.7	53.2	31.5
DEF	62.2	50.6	55.0	50.8	47.3	35.2
IV	65.5	56.9	62.4	55.0	45.8	38.3
IPR	63.0	55.1	59.6	56.3	49.3	39.3
LR	60.1	57.3	54.3	55.2	43.4	32.4
OCC	59.2	54.3	57.4	54.0	47.6	34.0
OPR	63.1	55.7	59.8	54.6	48.9	38.0
OV	55.0	50.6	55.3	42.3	43.5	31.7
SV	62.5	55.7	56.4	55.6	48.6	41.6

Table 3.
Success of Siamese tracker over different challenges.

and GOTURN failed to track the runner while SIAMRPN, SINT, and CNNSI have successfully tracked the runner. Meanwhile, the *ClifBar* sequence portrays SV, BC, MB, FM, IPR and OCC challenges and CNNSI only tracked the object efficiently while others failed. *FaceOcc1* and *Jogging-1* clearly show the occlusion challenge. We observe that all the trackers have tracked successfully face of the lady in *FaceOcc1* sequence where lady partially rotates a book in front of her face. While in *Jogging-1* sequence where occlusion is presented by a pole, all the Siamese trackers succeeded to track the lady except GOTURN. Another challenging sequence is *CarScale* which clearly shows that the size of the car is changing with the passage of time. We note that CFNet tracked the car efficiently while the rest of the trackers only tracked some region of the car.

4.3 Speed analysis

We also reported the speed of the trackers as frames per second (fps) as shown in **Figure 12**. We observe that GOTURN is computational cost effective and

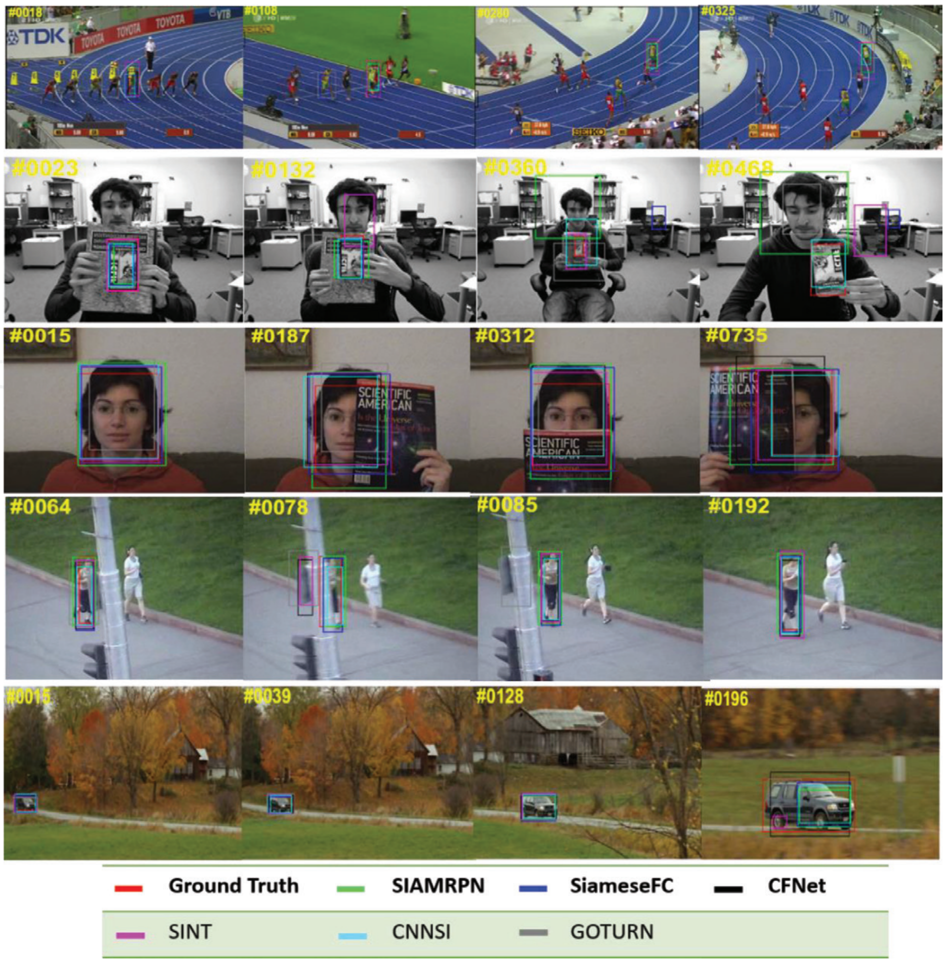

Figure 11.
Qualitative analysis of Siamese trackers over Bolt, ClifBar, FaceOcc1, Jogging-1, and CarScale sequences.

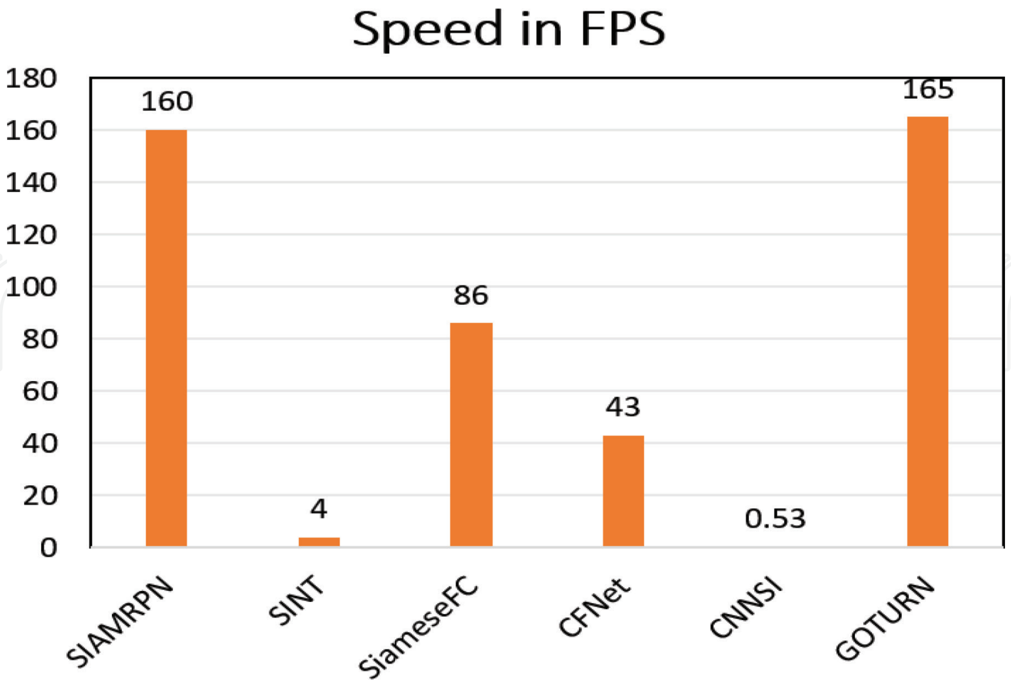

Figure 12.
Speed analysis of the trackers.

tracks objects at a speed of 165 fps. Similarly, SIAMPRN is also computational cost-efficient and can track at 160 fps. Although SiameseFC and CFNet have high computational cost compared to GOTURN and SIAMRPN but still manage to track at high speed. However, SINT (4 fps) and CNNSI (0.53 fps) have very low speed and consume a lot of computational costs.

5. Summary of Siamese networks comparison

We study three different types of Siamese network architectures employed in visual tracking application. We observe that all the Siamese trackers exploit the discriminative ability of deep CNN features. Experimental study revealed that late merge technique is better than others. **Table 4** shows the characteristics of the different architecture of Siamese networks.

	Late merge	Intermediate merge	Early merge
Definition	Inputs are combined at the final layer	Inputs are combined well before the final layer	Inputs are stacked before feeding network
Trackers	SiameseFC, CFNet, SINT, SIAMRPN	GOTURN, YCNN, EAST	CNNSI, SiameseCNN
Output (bounding box/ score map/ scores)	All	All	Scores
Features exploitation	Exploits the input images separately which are more discriminative	Initially exploits the input images features and then fused features are exploited which reduces the discriminative ability	Inputs are merged and then processed which reduces the discriminative ability of deep CNN features
Performance (precision and success)	Efficient	Moderate	Moderate
Speed	Fast	Fast	Slow

Table 4.
Characteristics of Siamese trackers.

6. Conclusions and future directions

In this chapter we study Siamese networks and their different variants for the task of visual object tracking. Siamese networks are classified into three categories based on their architecture including late merge, intermediate merge, and early merge. We observe that late merge Siamese trackers have shown better performance compared to the other trackers. Our study concludes that SIAMRPN has shown outstanding performance and ranked the best among the selected Siamese trackers. The tracking performance of the Siamese trackers can be improved by integrating both the spatial and temporal information. We observe that almost all the Siamese Networks do not perform the online model update. It would be a great challenge to update the model during the tracking while maintaining the robustness of the Siamese trackers. Other deep features such as RNN, Residual Net and GAN can be exploited within the Siamese networks to improve the tracking performance. Zero-shot and one-shot learning are getting popular due to the limited data issue. Integration of zero-shot and one-shot with Siamese trackers is yet to be explored in the visual object tracking field.

Acknowledgements

This research was supported by Development project of leading technology for future vehicle of the business of Daegu metropolitan city (No. 20190405).

Author details

Mustansar Fiaz[1], Arif Mahmood[2] and Soon Ki Jung[1]*

1 Kyungpook National University, Daegu, Republic of Korea

2 Information Technology University, Lahore, Pakistan

*Address all correspondence to: skjung@knu.ac.kr

References

[1] Laurense VA, Goh JY, Gerdes JC, editors. Path-tracking for autonomous vehicles at the limit of friction. In: 2017 American Control Conference (ACC); IEEE. 2017

[2] Brown M, Funke J, Erlien S, Gerdes JC. Safe driving envelopes for path tracking in autonomous vehicles. Control Engineering Practice. 2017;**61**:307-316

[3] Ali A, Jalil A, Niu J, Zhao X, Rathore S, Ahmed J, et al. Visual object tracking—Classical and contemporary approaches. Frontiers of Computer Science. 2016;**10**(1):167-188

[4] Liu G, Liu S, Muhammad K, Sangaiah AK, Doctor F. Object tracking in vary lighting conditions for fog based intelligent surveillance of public spaces. IEEE Access. 2018;**6**:29283-29296

[5] Tian B, Yao Q, Gu Y, Wang K, Li Y, editors. Video processing techniques for traffic flow monitoring: A survey. In: 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC); IEEE. 2011

[6] Datondji SRE, Dupuis Y, Subirats P, Vasseur P. A survey of vision-based traffic monitoring of road intersections. IEEE Transactions on Intelligent Transportation Systems. 2016;**17**(10):2681-2698

[7] Rautaray SS, Agrawal A. Vision based hand gesture recognition for human computer interaction: A survey. Artificial Intelligence Review. 2015;**43**(1):1-54

[8] Maqueda AI, del-Blanco CR, Jaureguizar F, García N. Human-computer interaction based on visual hand-gesture recognition using volumetric spatiograms of local binary patterns. Computer Vision and Image Understanding. 2015;**141**:126-137

[9] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv: 14091556; 2014

[10] Schuster M, Paliwal KK. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing. 1997;**45**(11):2673-2681

[11] Kodirov E, Xiang T, Gong S, editors. Semantic autoencoder for zero-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017

[12] He K, Zhang X, Ren S, Sun J, editors. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016

[13] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al., editors. Generative adversarial nets. In: Advances in Neural Information Processing Systems. 2014

[14] Chen J-C, Patel VM, Chellappa R, editors. Unconstrained face verification using deep CNN features. In: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV); IEEE. 2016

[15] Wang J, Yang Y, Mao J, Huang Z, Huang C, Xu W, editors. CNN-RNN: A unified framework for multi-label image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016

[16] Girshick R, Donahue J, Darrell T, Malik J, editors. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014

[17] Moeskops P, Wolterink JM, van der Velden BH, Gilhuijs KG, Leiner T, Viergever MA, et al., editors. Deep

learning for multi-task medical image segmentation in multiple modalities. In: International Conference on Medical Image Computing and Computer-Assisted Intervention; Springer. 2016

[18] Ren S, He K, Girshick R, Sun J, editors. Faster r-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems. 2015

[19] Fiaz M, Mahmood A, Javed S, Jung SK. Handcrafted and Deep trackers: Recent Visual Tracking Trends and Approaches. ACM Computing Surveys; 2019

[20] Krizhevsky A, Sutskever I, Hinton GE, editors. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. 2012

[21] Wu Y, Lim J, Yang M-H, editors. Online object tracking: A benchmark. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2013

[22] Wu Y, Lim J, Yang M-H. Object tracking benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2015;**37**(9):1834-1848

[23] Li P, Wang D, Wang L, Lu H. Deep visual tracking: Review and experimental comparison. Pattern Recognition. 2018;**76**:323-338

[24] Leang I, Herbin S, Girard B, Droulez J. On-line fusion of trackers for single-object tracking. Pattern Recognition. 2018;**74**:459-473

[25] Zhang S, Yao H, Sun X, Lu X. Sparse coding based visual tracking: Review and experimental comparison. Pattern Recognition. 2013;**46**(7):1772-1788

[26] Yang M, Wu Y, Hua G. Context-aware visual tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2009;**31**(7):1195-1209

[27] Babenko B, Yang M-H, Belongie S. Robust object tracking with online multiple instance learning. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2011;**33**(8):1619-1632

[28] Hare S, Golodetz S, Saffari A, Vineet V, Cheng M-M, Hicks SL, et al. Struck: Structured output tracking with kernels. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2016;**38**(10):2096-2109

[29] Zhang K, Zhang L, Yang M-H, editors. Real-time compressive tracking. In: European Conference on Computer Vision; Springer. 2012

[30] Bolme DS, Beveridge JR, Draper BA, Lui YM, editors. Visual object tracking using adaptive correlation filters. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition; IEEE. 2010

[31] Henriques JF, Caseiro R, Martins P, Batista J. High-speed tracking with kernelized correlation filters. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2015;**37**(3):583-596

[32] Danelljan M, Hager G, Shahbaz Khan F. Learning spatially regularized correlation filters for visual tracking. In: Felsberg M, editor. Proceedings of the IEEE International Conference on Computer Vision. 2015

[33] Danelljan M, Hager G, Shahbaz Khan F, Felsberg M, editors. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016

[34] Li F, Tian C, Zuo W, Zhang L, Yang M-H, editors. Learning spatial-temporal

regularized correlation filters for visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018

[35] Lukezic A, Vojir T, Čehovin Zajc L, Matas J, Kristan M, editors. Discriminative correlation filter with channel and spatial reliability. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017

[36] Nam H, Han B, editors. Learning multi-domain convolutional neural networks for visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016

[37] Nam H, Baek M, Han B. Modeling and propagating cnns in a tree structure for visual tracking. arXiv preprint arXiv: 160807242; 2016

[38] Ma C, Huang J-B, Yang X. Hierarchical convolutional features for visual tracking. In: Yang M-H, editor. Proceedings of the IEEE International Conference on Computer Vision. 2015

[39] Qi Y, Zhang S, Qin L, Yao H, Huang Q, Lim J, et al., editors. Hedged deep tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016

[40] Briechle K, Hanebeck UD, editors. Template matching using fast normalized cross correlation. In: Optical Pattern Recognition XII; International Society for Optics and Photonics. 2001

[41] Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2012;**34**(7):1409-1422

[42] Wang S, Lu H, Yang F, Yang M-H, editors. Superpixel tracking. In: 2011 International Conference on Computer Vision; IEEE. 2011

[43] Nguyen HT, Smeulders AW. Robust tracking using foreground-background texture discrimination. International Journal of Computer Vision. 2006;**69**(3):277-293

[44] Godec M, Roth PM, Bischof H. Hough-based tracking of non-rigid objects. Computer Vision and Image Understanding. 2013;**117**(10): 1245-1256

[45] Held D, Thrun S, Savarese S, editors. Learning to track at 100 fps with deep regression networks. In: European Conference on Computer Vision; Springer. 2016

[46] Bertinetto L, Valmadre J, Henriques JF, Vedaldi A, Torr PH, editors. Fully-convolutional siamese networks for object tracking. In: European Conference on Computer Vision; Springer. 2016

[47] Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Fei-Fei L, editors. Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014

[48] Zbontar J, LeCun Y, editors. Computing the stereo matching cost with a convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015

[49] Schroff F, Kalenichenko D, Philbin J, editors. Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015

[50] Zagoruyko S, Komodakis N, editors. Learning to compare image patches via convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015

[51] Tao R, Gavves E, Smeulders AW, editors. Siamese instance search for tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016

[52] Valmadre J, Bertinetto L, Henriques J, Vedaldi A, Torr PH, editors. End-to-end representation learning for correlation filter based tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017

[53] Li B, Yan J, Wu W, Zhu Z, Hu X, editors. High performance visual tracking with siamese region proposal network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018

[54] Chen K, Tao W. Once for all: A two-flow convolutional neural network for visual tracking. IEEE Transactions on Circuits and Systems for Video Technology. 2018;**28**(12):3377-3386

[55] Huang C, Lucey S. Learning policies for adaptive tracking with deep feature cascades. In: Ramanan D, editor. Proceedings of the IEEE International Conference on Computer Vision. 2017

[56] Fiaz M, Mahmood A, Jung SK. Convolutional neural network with structural input for visual object tracking. In: ACM Symposium on Applied Computing. 2019

[57] Leal-Taixé L, Canton-Ferrer C, Schindler K, editors. Learning by tracking: Siamese CNN for robust target association. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2016