ALTO: ADVERSARIAL LEARNING FOR TRACKING OBJECTS

submitted in partial fulfilment of the requirements

for the degree of

BACHELOR OF TECHNOLOGY

in

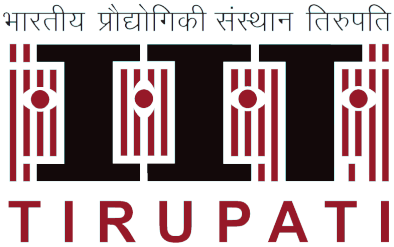
ELECTRICAL ENGINEERING

by

BALA SURAJ PEDASINGU EE15B017

Supervisor

DR. RAMA KRISHNA SAI GORTHI



DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY TIRUPATI

MAY 2019

ACKNOWLEDGEMENTS

I am grateful to my **Parents** for enriching my world with their support and encouragement during the course of my B.Tech program.

I have taken sincere efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to **Dr. RamaKrishna Sai Gorthi** for his guidance and constant supervision as well as for providing necessary information regarding the project and also for their support in encouraging to complete the project.

I would like to express my gratitude towards **Mr.Litu Rout** for his kind co-operation and encouragement in nuturing my idea of this project to make it successful and deployable. I would like to express my special gratitude and thanks to **Mr. P Naveen** for contributing to this project with constant attention and time.

My sincere thanks and appreciations also goes to the people who have willingly helped me out with their technical abilities in developing the project

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Visual object tracking is a classical problem of estimating the trajectory of an target object in a video sequence, provided its location in the first frame. It is one of the fundamental problems in computer vision and can serve as a building block in complex vision systems. There are wide range of practical applications such as automatic surveillance (Huang and Fu (2011)), autonomous driving (Chen *et al.* (2015)), Medical vision systems (Arvind *et al.* (2016)) and Video analysis. This thesis will deal with the problem of improving the accuracy of state-of-the-art Siamese family trackers.

1.1 Motivation and Background

Visual object tracking is one of the classical problems in computer vision, with a vast number of practical applications. In a complex video sequence with the target state i.e. bounding box in the first frame, a tracker must estimate the target states in all successive frames. It is a quite challenging task by various reasons. Firstly, a tracker must track objects belonging to any arbitrary class, which is quite contrary to computer vision tasks such as classification and detection in which the objects belongs to a known set of classes. Secondly, the target object can undergo complex appearance changes due to cluttered background, motion blur, deformations, occlusion and illumination variation. Therefore, there is great necessity for tracker to learn a highly-robust appearance model of the target object to handle these challenges from the limited information in a single frame.

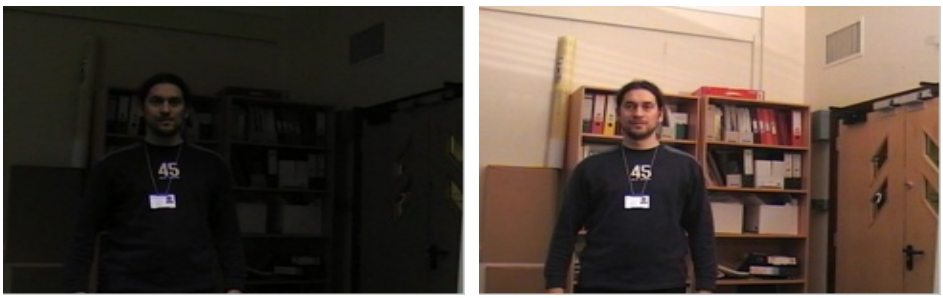
The richness of the object tracker is determined using two criterion i.e. robustness and accuracy. Robustness is a temporal aspect which is the ability of a tracker to track for long duration, without drifting from the target. To achieve high robustness, a tracker must reliably distinguish the target from the background and other distracting objects. Accuracy is a spatial aspect, which quantifies how accurately a tracker can estimate the bounding box. To achieve high accuracy, a tracker must precisely estimate the object boundary in the search image.

The task of tracking can be categorized into two sub-blocks, one block deals with robustness namely target classification, and another block deals with the accuracy namely target estimation. In target classification, a reliable coarse location of the target in the image must be provided by classifying image patches into foreground and background. To achieve robust target classification, a tracker should learn a target model invariant to changes in the target appearance caused by brightness changes, deformations, background clutter, motion blur etc. In target estimation, a precise estimation of the target state, often represented by a bounding box is required to achieve high tracking accuracy. It is a quite challenging problem since the tracker needs high-level knowledge about the target object to determine its state. In cases of deformations and out-of-plane rotations, the tracker should have information that constitutes an object boundary to determine the bounding-box coordinates.

In recent years, the attention of tracking communities has been invested into constructing robust classifiers, based on correlation filters (Valmadre *et al.* (2017) and Danelljan *et al.* (2015*a*)) and exploiting powerful deep feature representations (Gundogdu and Alatan (2018) and Bertinetto *et al.* (2016)). On contrary, focus on target estimation is below expected progress. In fact, most state-of-the-art trackers still rely on the target classification for target estimation by performing a multiscale search. However, this strategy is naive and fundamentally limited since bounding box estimation is inherently a challenging task, requiring high-level understanding of the object's posture. It highlights the necessity for a separate component for target estimation to handle changes in target state.

1.2 Prominent Challenges

- **Photometric Transformations**
  - **Illumination changes:** It has adverse effects on target appearance model since color features, edges and object contours are sensitive to illumination changes as shown in Fig. 1.1



(a) (b)

Figure 1.1: Image sequence with Illumination variation.

- **Geometric Transformations**
  - **Scale changes:** Target object appears at various scales due to in-out motion from camera's field. Therefore, object information lies in coarse-fine range at different instances as shown in Fig. 1.2

(a) (b)

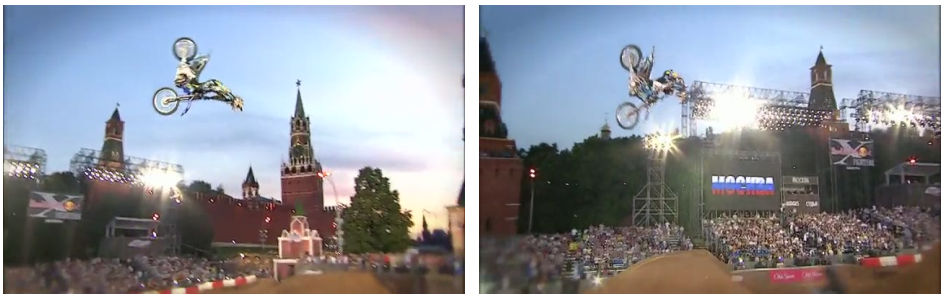
Figure 1.2: Image sequence with scale variation.

  - **Shape changes:** Target objects appear differently by undergoing various deformations, which leads to different model representations and eventually target state gets drifted as shown in Fig. 1.3

(a) (b)

Figure 1.3: Image sequence with shape variation.

– **Rotation changes:** It is a fundamental problem to detect different rotated versions since traditional features are sensitive to rotations as shown in 1.4



(a) (b)

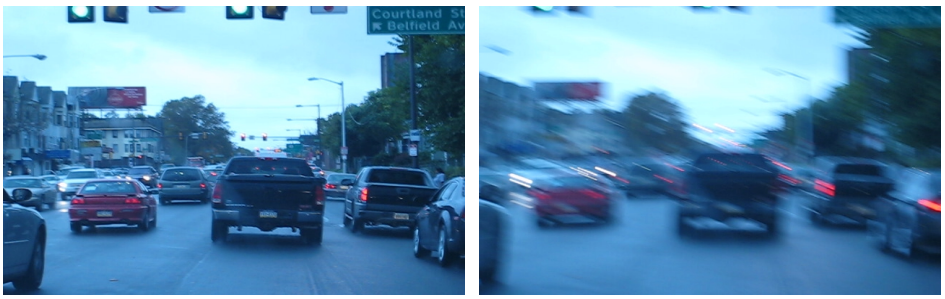
Figure 1.4: Image sequence with rotation variation.

- **Background clutter** : The target state estimation becomes tedious in cluttered background since edges and counters are less distinguishable as shown in Fig.1.5

(a) (b)

Figure 1.5: Image sequence with Background clutter.

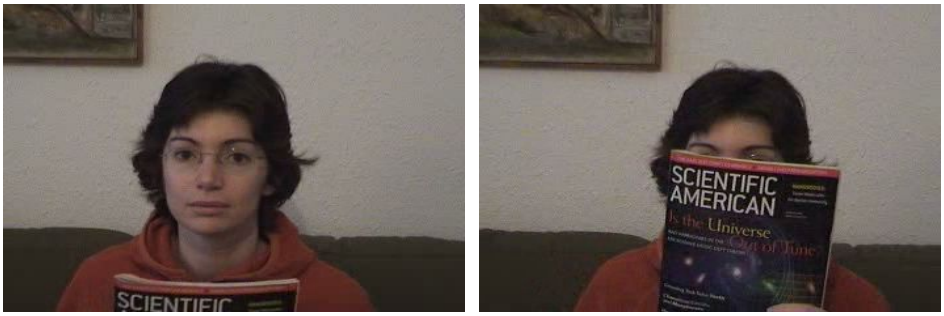
- **Motion blur** : In constant motion of camera/object, image captured gets smudged leading to loss of boundary details and fine texture patterns as shown in Fig. 1.6.



(a) (b)

Figure 1.6: Image sequence with Motion-Blur.

- **Occlusion** : In certain scenarios, object is partially clouded by obstacles, leading ambiguous model representations for drifting as shown in Fig. 1.7.



(a) (b)

Figure 1.7: Image sequence with Occlusion.

1.3 Practical Applications

- **Autonomous Driving** : It is a crucial component in concrete reality to pave the way for future systems. Autonomous driving is generically multi-object tracking task in which proper estimation of several object trajectories is essential for optimizing actual short and safe trajectory between source and destination for navigation. Information processed from tracking has high importance in fusion with information from various sensors to determine navigation paths. There is a visual example of autonomous cars in streets as shown in Fig. 1.10.

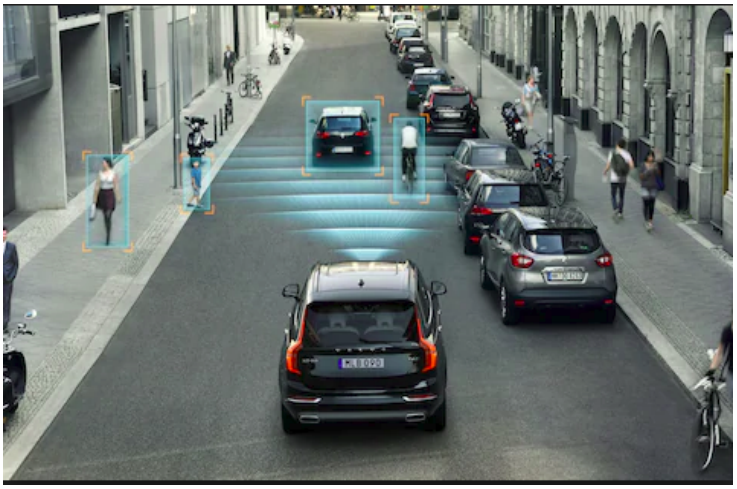


Figure 1.8: AI based Autonomous cars.

- **Medical Imaging** : In medical applications such as blood sample analysis, percentage estimation of various cells by tracking provide crucial information for determining patient's condition. In medical operations such as Endoscopy, there is necessity for monitoring the motion of various cells inside human body.

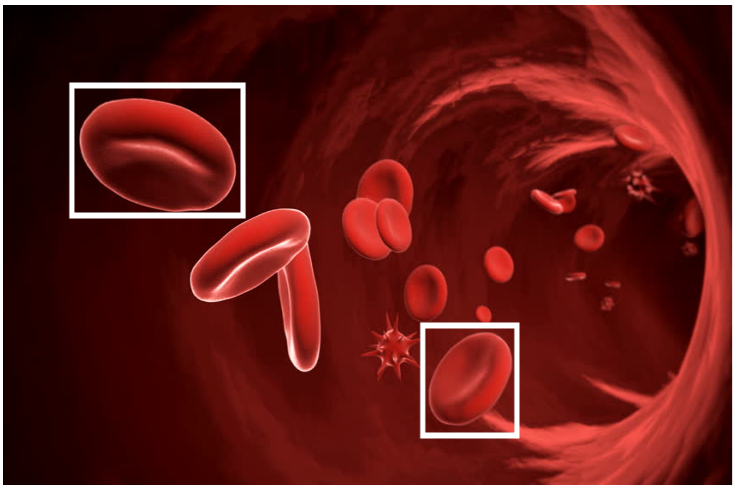


Figure 1.9: Tracking Red blood cells.

- **Autonomous Surveillance** : In modern day organizations, activity monitoring and anomaly detection are foremost concerning aspects to tackle with. Anomaly detection is identification of unusual patterns/signals in events/observations which don't conform with expected behavior causing suspicion.

  It has many applications in business such as intrusion detection by identifying strange packets in network traffic that signal a hack, health monitoring systems for spotting a malignant tumor in an MRI scan, and fraud detection in credit card transactions. Activity monitoring is identification of certain known set of actions in observation environments with applications such as monitoring babies, student activities in schools etc.

Figure 1.10: Surveillance for Activity monitoring.

1.4 Research Objectives

- **To explore the role of adversarial learning in improving the power of deep convolution embedding networks to enhance tracking accuracy .**
  - To improve the foreground-background discriminative capabilities of siamese networks through harnessing the generative power from adversarial learning.
  - To design a novel end-to-end learning framework for tracking by combining two powerful breakthroughs Siamese networks and Generative Adversarial Networks (GANs) for balanced accuracy and speed.
- **To utilize the backbones in various deep CNN architectures by investigating the control aspects governing the tracking accuracy and robustness.**
  - To harness power of residual module in deep CNNs for creating powerful embeddings invariant of changes in object appearances.
  - To ensemble similarity learning and dissimilarity learning network and get combined response for better localization precision.
  - To improve target localization by combined decision of different spatial resolution responses.

1.5 Contribution of the thesis

- We propose a novel end to end framework based on adversarial learning for object tracking. This framework enhances prediction of the target localization in learning based trackers through adversarial learning, by minimizing error between the predictions and ground truth. Our experiments show that the proposed approach improves the baseline trackers and outperforms with state-of-the-art methods on well known benchmark datasets.
- We built the ensemble model of similarity and dissimilarity learning network, which is optimized at two-stages of the network where different spatial-resolution embeddings are generated. Our experiments on various architectures outperforms tracking performance with various state-of-art methods.

1.6 Outline of the Thesis

The thesis has been presented in a number of chapters with different aspects. The rest of the thesis is organized as follows. Chapter 2 provides Literature review about Visual object tracking and describes the evolution of different families based on approach methodology and categorization of trackers based on their performance. Then, it describes the objective of visual object tracking in modern-era. Chapter 3 starts with detailed description of the baseline approach along with certain newly emerged state-of-art learning methods in deep learning. Finally, it describes detailed explanation of proposed framework and various experimentation. Chapter 4 describes understanding of evaluation metrics and provides some information regarding standard benchmark datasets. Chapter 5 provides experimental evaluations and comparisons with state-or-art methods in terms of qualitative metrics along with visual results for intuitive understanding.

CHAPTER 2

LITERATURE REVIEW

2.1 Brief history of tracking

Object Tracking can be defined as the classical problem of estimating the object path in the image plane in a given scene. The aim of an object tracking is to generate the path for an object above time by finding its position in every single video frame. In a scene, Object detection and determining correspondence between the object occurrences through frames can be accomplished in independent manner or sequential manner. In the former stage, Region of interest in given frame is achieved through object detection algorithms, and then tracking corresponds to objects across frames. In the later stage, the object region is projected by sequentially updating object state i.e. location and size, obtained from previous frames. In the survey of object tracking Parekh *et al.* (2014), the author describes evolution of visual object tracking, where it started being classified into point tracking, kernel based tracking and silhouette based tracking.

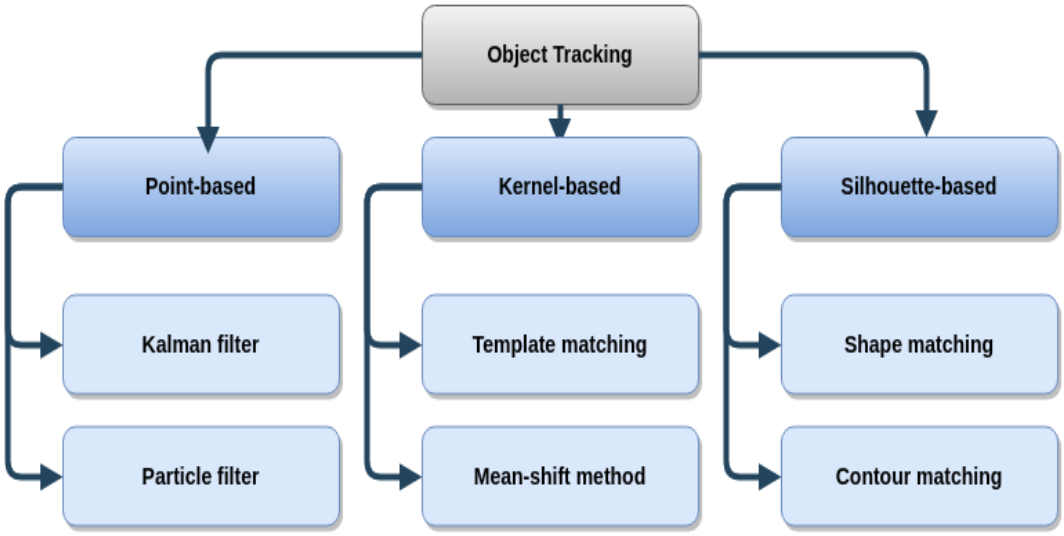


Figure 2.1: Various types of Object tracking

2.1.1 Point based tracking

In the structure of a image, target objects are represented by their feature key-points during tracking. Point tracking is a complex problem especially in the scenarios where occlusions, false detections are quite possible. Recognition can be accomplished by thresholding, on identification of these points. These methods use qualitative motion heuristics to constrain the correspondence problem.

Kalman Filtering

In Optimization algorithms by Recursive Data Processing, The Kalman Filter conduct the restrictive probability density propagation. It is a list of mathematical equations that provides an efficient computational platform to estimate the state of a process in various aspects. It predicts the estimations of past, present, and future states, even when the characteristic nature of the modelled system is unknown. The Kalman filter estimates a process by using a means of control feedback. The filter estimates the process state at given time and obtains feedback signals corrupted with noise. The kalman equations can be classifies as time update equations and measurement update equations. The former set of equations are responsible for propagating forward in time, the current state and error co-variance estimates to obtain the prior estimate for the next state. The measurement update equations are responsible for the control feedback. It always give optimal solutions.

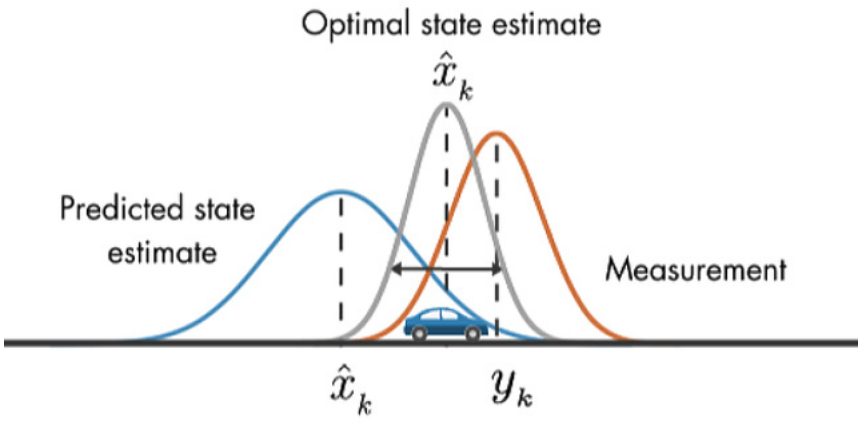


Figure 2.2: Intuitive description of Kalman filter [source]

Particle Filtering

The particle filter generates all possible states of the models for one variable before transitioning to next variable. This algorithm has a upper-hand when variables are generated dynamically. It also allows for re-sampling operation. State variables are Gaussian distributed, is a limitation of the Kalman filter. So, it gives poor approximations of state variables which don't fall under Normal distribution. This limitation can be bypassed by using particle filtering. This algorithm uses contours maps, color features, or texture patterns.

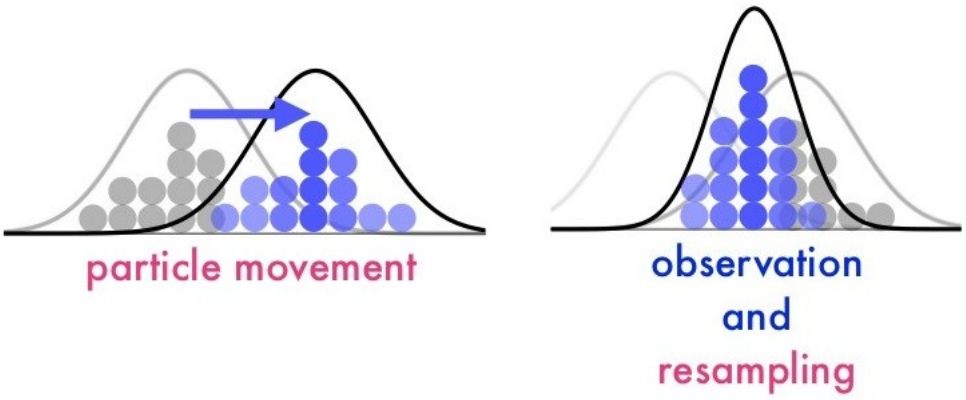


Figure 2.3: Intuitive description of Particle filter [source]

2.1.2 Kernel based tracking

Kernel tracking is generically conducted by computing the moving object, which is represented by a object state or region in sequential frames. The object motion has parametric form such as translation, affine transformation etc. These algorithms differ in terms of the methods of target representations, the number of tracking objects, and the approximating method for the object motion. In real-time, exemplification of object using geometric figures is generic.They are wide range of tracking techniques based on object representation, object features and object appearance.

Template Matching

Template matching is a trail-and-error method for determining the region of interest. In template matching, a template image is verified with the other frame in the video. Template matching is a iterative technique to process images to find image patches that matches with target template in each frame. The procedure for matching can be accomplished using distance metrics, similarity measures etc.

Figure 2.4: Template Matching [source]

Mean Shift tracking

Mean-shift tracking performs exhaustive search to localize the area in the frame which is most similar to a previous initialized model. The region of image for tracking is represented by a histogram. The gradient ascent is applied to move the tracker to the location that maximizes a similarity response between the model and current search region. In object tracking algorithms, target representation is usually bounding box (rectangular, elliptical or polygonal). To characterize the target candidate, histogram of color-space intensities is chosen. Finally, the model is represented by its probability density function and the model is regularized with an asymmetric kernel.

2.1.3 Silhouette based tracking

Certain objects have complex irregular shapes which cannot be well-defined by simple geometric states. Silhouette based techniques are capable of describing these objects. The aim of a silhouette-based object tracking is determining regions of interest for object in every frame by object model generated by the previous frames silhouettes.

Shape matching

Shape matching can be functional tracking in which an object silhouette and its associated model are searched in the current frame, which is analogous to tracking based on template matching. In this approach, the exhaustive search is performed by computing the similarity score of the object with the model generated from the presume object silhouette based on previous frame.

Contour matching

In contrary to shape matching, contour tracking method is iterative evolution of an initial contour in the previous frame to its new location in the current frame. The progress of this evolution requires the part of the object in the present frame to overlap with the object area in the past frame. This evolution process can be performed using two different methods. The former method uses state space models to model the contour and its motion. The later approach evolves the contour by minimizing the energy of the contour using optimization techniques such as gradient descent.

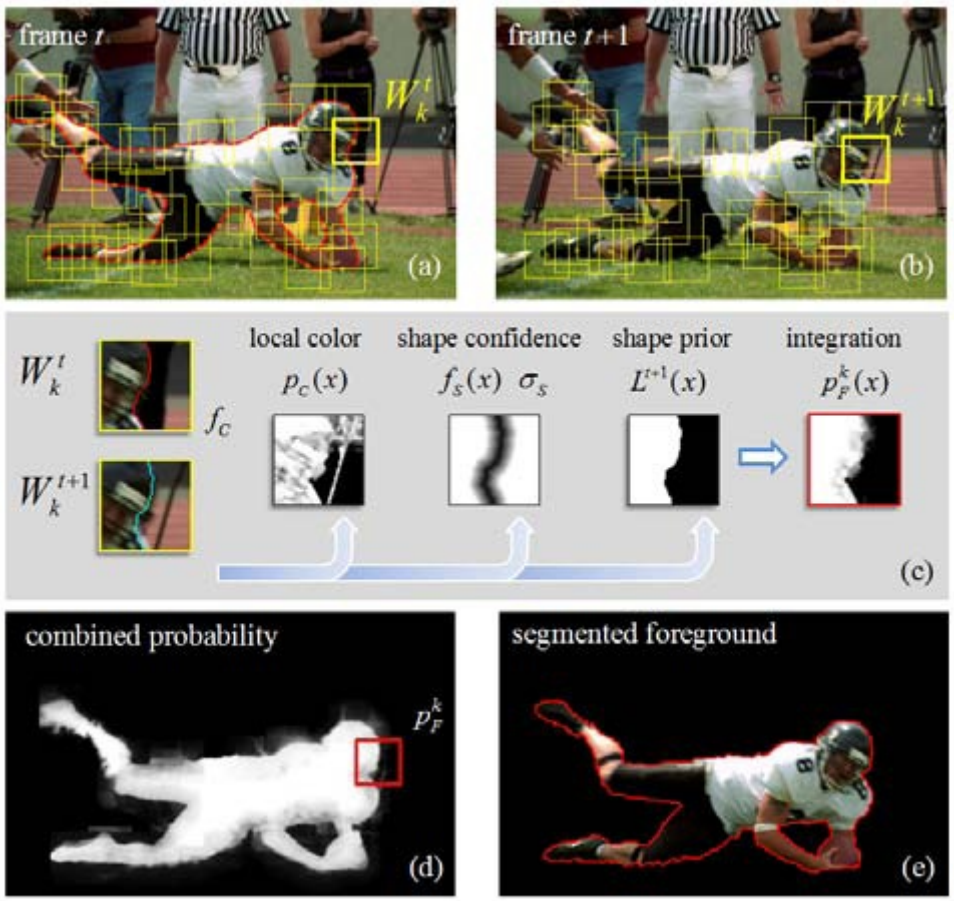


Figure 2.5: Contour Matching [source]

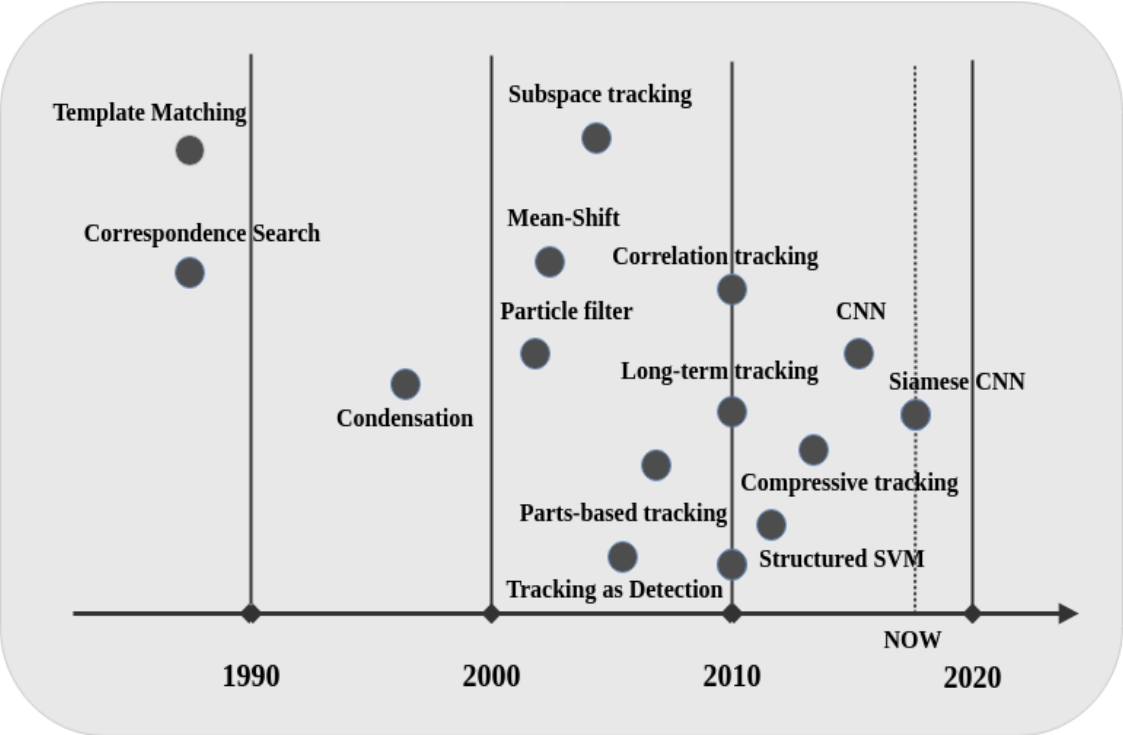


Figure 2.6: Brief history of Object tracking [source]

2.2 Tracking formulation in Modern-era

In recent times, Object tracking has been classified into two broad classes depending on their real-time robustness i.e. Short-term tracking and Long-term tracking. The task of tracking can be broken down into two independent sub-problems. These sub-problems are quite challenging in different ways. The first sub-problem, which call **Target classification**, the tracker must learn an target appearance model to discriminate the target from the background, while being invariant to changes in target appearance caused by rotations, deformation, motion blur etc. The second task, which we call **Target estimation**, the tracker must know what constitutes the object boundary such that it can determine a bounding box which fits the object perfectly.

2.2.1 Target Classification

In recent years, Discriminative Correlation Filters (DCF) [Bolme *et al.* (2010), Danelljan *et al.* (2017), Kiani Galoogahi *et al.* (2017)] have gained significant attention to tackle the problem of target classification and attain state-of-the-art results. In these methods, an

correlation filter is online-trained discriminatively to model the appearance of the target in the first frame. The model is trained using the image patches containing the target as positive training samples and the background patches as negative training samples. The basic strategy utilizes all the shifted versions of the image for training, making the problem to be diagonalized in mathematical sense and solved efficiently in the Fourier domain. The early methods used basic training methods to operate at high frame rates (100 fps). On other hand, a variety of improvements have been proposed to make the classifier robust. SRDCF [Danelljan *et al.* (2015*b*)] introduces a spatial regularization term in the classifier learning to penalizes the filter coefficients based on their spatial location. It enables the classifier to be trained using a larger image region, using more training data. BACF [Kiani Galoogahi *et al.* (2017)] also proposed to utilize additional training data by sampling negative examples from the complete image, in contrary to standard DCF, which uses circular-shifted patches around the target. CSRDCF [Lukezic *et al.* (2017)] learns features spatially in different channels which helps the filter to focus on object region. CCOT [Danelljan *et al.* (2016)] trains a filter in the continuous domain along with usage of features with different resolutions, while providing sub-pixel accuracy. Then, ECO [Danelljan *et al.* (2017)] introduces a factorized convolution operator to facilitate dimensional reduction to provide a drastic increase in tracking speed while reducing model over-fitting.

In development of robust learning methods, a important factor that led to the improvements in DCFs comes with using powerful features. Initially, grayscale intensity values were used as input features for the appearance model [Bolme *et al.* (2010)]. Later, hand-crafted features such as histogram-of-gradients (HOG) [Dalal and Triggs (2005)] and color names (CN) [Van De Weijer *et al.* (2009)] were employed. Using the features extracted from deep layers of a convolutional neural network (CNN) [Danelljan *et al.* (2016), Ma *et al.* (2015), Sun *et al.* (2018)] become the recent trend. These high-dimensional features exhibit significant invariance to appearance changes in the target, compared to the hand-crafted features to give higher robustness. Then emerges a new family of target classifiers, which become popular in recent years, for high speed tracking are Siamese methods [Bertinetto *et al.* (2016), Li *et al.* (2018), Tao *et al.* (2016)]. They train a network to predict the similarity score between two images. Images patches

with same object from different frames in a sequence must achieve high similarity score, while giving low scores for the images patches of different objects. This network can be trained offline using large sequences of video datasets. During online tracking, the target is identified in each frame through correlation measure. There is no need for training appearance model, leading to high tracking speeds. RASNet [Wang *et al.* (2018)] attempts to solve distraction problem by predicting residual attentions to weight the spatial regions and channels having most discriminative features. SiamRPN [Li *et al.* (2018)] uses a region proposal network (RPN) [Ren *et al.* (2015)], in pipeline with siameseFC network. It classifies each image patch into target or background. The filter weights are predicted by template branch of the siamese network, using the first frame. DaSiamRPN [Zhu *et al.* (2018)] further improves SiamRPN by performing distraction aware training, i.e. explicitly training the network to handle distractors. In addition, MDNet [Nam and Han (2016)] trains a Convolutional Neural Network (CNN) binary classifier for tracking. The network contain several shared layers and domain specific layers. The shared layers are trained offline, using many videos to learn general representation for target classification. The domain specific layers are trained online for a particular video sequence. GOTURN [Held *et al.* (2016)] regresses the target bounding box given the current frame, and a patch centered at the target from the previous frame.

2.2.2 Target Estimation

In contrary to target classification, target estimation has received less attention in the recent years. Most current state-of-the-art trackers [Bhat *et al.* (2018), Danelljan *et al.* (2017), Sun *et al.* (2018)] don't have a separate component for target estimation. Instead, they estimate target through a naive way of performing a multi-scale search, which depends on the classification component i.e. the cropped image is re-sized to several different scales. The location and the scale having the highest score is selected as target state. There are two major drawbacks in this approach. Firstly, it models simple scaling of the target only. Aspect ratio changes cannot be handled by this approach. Secondly, this results in tuning a certain set of hyper-parameters, which are specific to datasets. As a result, the generic nature of the framework is lost. A target estimation approach which

is widely used is the discriminative scale space tracker (DSST). It trains an explicit filter online for scale estimation, by sampling the target at different scales. However, the training data contains only scaled versions of the target but not changes in target size caused by deformation.

A few approaches have utilized bounding box regression, commonly used in object detection. MDNet [Nam and Han (2016)] trains a bounding box regressor to predict the bounding box coordinates, using the groundtruth annotation in the first frame. Unlike DSST, this approach can predict boxes with different aspect ratios. However, since the regressor is trained on a single image, it suffers from the lack of diversity in data. SiamRPN [Li *et al.* (2018)], as discussed before, uses a region proposal network (RPN) in which one branch predicts the offsets to be added to the anchor box to get the bounding box for the target. The network is trained offline end-to-end using large scale video datasets. Thus, it can learn a general representation for various objects. The drawback of this approach is the target estimation is closely coupled with target classification.

CHAPTER 3

IMPLEMENTATION FRAMEWORK

3.1 Baseline approach: Siamese Networks

In development of the discriminative correlation methods, a new emerged approach of similarity learning showed a different way of learning to track arbitrary objects. In this approach, a fully convolutional function *F(z, x)* that compares an exemplar image z to a candidate image x of the same size and returns a high response score if the two images depict the same object and otherwise a low score. To determine the position of the object in a new image, it performs exhaustive search over all possible locations and choose the target candidate with the maximum similarity score to the previous appearance of the object. In these approach, it simply uses the initial appearance of the object as the template. The function F will be learned from a large scale video dataset with annotated object trajectories.

Due to its widespread success in computer vision [Sharif Razavian *et al.* (2014), Parkhi *et al.* (2015)], this approach uses a deep CNN as the function F. Similarity learning with deep CNNs is typically taken care by using Siamese architectures [18,19]. Siamese networks use an identical transformation θ to both inputs and then combine their representations using another function G according to $F(z, x) = G(\theta(z), \theta(x))$. The function G is a similarity metric, the function θ can be denoted as an embedding function. In past scenario, Deep siamese CNNs have been applied to tasks such as face verification [Taigman *et al.* (2014)], keypoint descriptor learning [Zagoruyko and Komodakis (2015)] and one-shot character recognition [Koch *et al.* (2015)].

They propose a fully-convolutional siamese architecture with respect to the candidate image x. since it commutes with translation. To define FC property, Let L_t denote the translation operator h(L_{tx})[u] = x[u - t], a function h maps inputs to outputs is

fully-convolutional with integer stride k for any translation t if

$$h(L_{kt}x) = L_t(h(x)) \tag{3.1}$$

In a fully-convolutional network, we can provide as input to the network any sized search image and it will compute the similarity at all translated subpatches on a dense grid in one evaluation. To accomplish this, we need a convolutional embeddings using function θ and cross-correlate the resulting feature maps to generate the response map.

$$F(z,x) = \theta(z) * \theta(x) + b1 \tag{3.2}$$

where b 1 denotes a signal which takes value b $\in$ R in every location. The output of this network is not a single score but rather a score map defined on a finite grid D $\subset Z^2$ as illustrated in Figure 3.1.

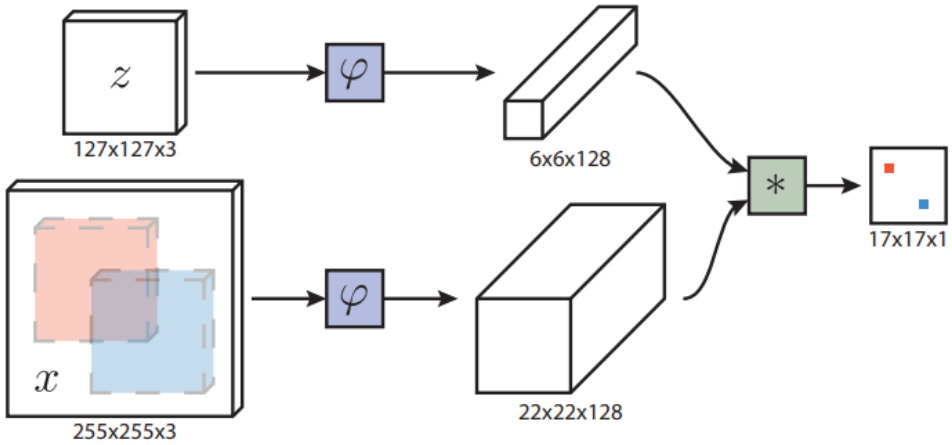


Figure 3.1: Fully-convolutional Siamese architecture (Bertinetto *et al.* (2016)). The score map is a scalar-valued whose dimension depends on the search image. In this figure, the red and blue pixels in the score map contain the similarities for the corresponding sub-windows.

3.1.1 Training Procedure

In discriminative training, train the network on positive and negative pairs by employing the logistic loss where v is the real-valued score of a single template-candidate pair and

y ∈ {+1, -1} is its ground-truth label.

$$\ell(y,v) = \log\left(1 + exp(-yv)\right) \tag{3.3}$$

It takes advantage of the fully-convolutional nature in network during training by using pairs that comprise a template image and a large search image to produce a score map $v : D \rightarrow R$, generating many samples per pair. It define the complete loss of a score map as the mean of the individual losses

$$L(y,v) = \frac{1}{|D|} \sum_{u \in D} \ell(y[u], v[u]) \tag{3.4}$$

Need a true label y[u] ∈ {+1, -1} for each position u ∈ D in the score map. The CNN parameters (θ) are obtained by applying Stochastic Gradient Descent (SGD)

$$\arg\min_{\theta} \mathop{\mathbb{E}}_{(z,x,y)} L(y, F(z,x \mid \theta) \tag{3.5}$$

The ground-truth of score map is designed in which element belongs to a positive example if it lies within radius R of the centre, considering the stride k of the network. The losses of the positive and negative examples in the score map are weighted to eliminate class imbalance.

$$y[u] = \begin{cases} +1, & \text{if} \quad \text{k } \|\text{u - v}\| \leq \text{R} \\ -1, & \text{otherwise} \end{cases} \tag{3.6}$$

Since the network is fully-convolutional, there is no problem in learning a bias for the sub-window at the centre. It presumes that search images centred on the target have the most difficult sub-windows and those adjacent patches to the target patch have the most influence on the performance of the tracker.

3.1.2 Tracking procedure

In online-tracking, in order to highlight fully-convolutional Siamese network and its generalization capability being trained on ImageNet large-scale video dataset, it employs an extremely simple algorithm in which the template embedding is generated from the first frame. Then, it performs exhaustive correlation over subsequent frame embeddings to determine the target location. In estimating target size, a naive multi-scale search method is employed to update the previous size, which is highly sensitive to target location. Despite its simplicity and naive approach, the tracking algorithm achieves good results on many benchmarks.

(a) (b) (c)

Figure 3.2: Tracking using SiameseFC framework.

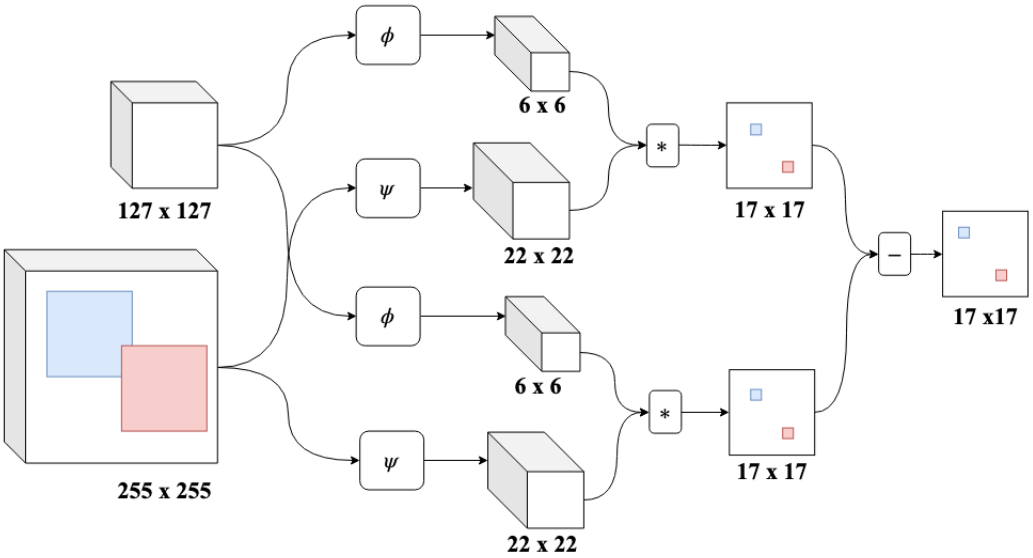
3.2 Experimentations based on Siamese networks:

3.2.1 Ensemble of Similarity and Dissimilarity network

In this experiment, we train two siamese networks(SN & DN) independently, in which one learns similarity between two image patches and vice-versa. In other words, it is a efficient way of designing correlation and anti-correlation blocks. In dissimilarity net, the ground-truth of score map is designed exactly opposite to siamese network. Then, the final response map is the combination obtained by subtracting DN score map from SN score map as shown in eq. 3.7

$$r[u] = \begin{cases} +2, & \text{High similarity} \\ 0 \pm \varepsilon & \text{Overlapping patches} \\ -2, & \text{Low similarity} \end{cases} \tag{3.7}$$

where r denotes the response map and ε is a small positive number.



ϕ - *Similarity net* | ψ - *Dissimilarity net*

Figure 3.3: Block diagram of ensemble of Similarity and dissimilarity networks

3.2.2 Fusion of coarse-fine response maps in Siamese network

Consider any version of siamese network, we must obtain two embeddings from the network for each image i.e. one embedding is collected from intermediate convolution layer till where effective stride's 4 and other embedding is obtained from end of the network having effective stride 8. Then, correlate the embeddings with effective stride 4, obtained from the template and search image to response maps of sizes 33. Similarly, we obtain score map with size 17 by correlating embeddings with effective stride 8.

$$y_1[u] = \begin{cases} +1, & \text{if} \quad 4 \, \|u - v\| \leq R \\ -1, & \text{otherwise} \end{cases} \quad \leftrightarrow \quad y_2[u] = \begin{cases} +1, & \text{if} \quad 8 \, \|u - v\| \leq R \\ -1, & \text{otherwise} \end{cases} \tag{3.8}$$

where y_1 and y_2 denotes coarse and fine score maps respectively.

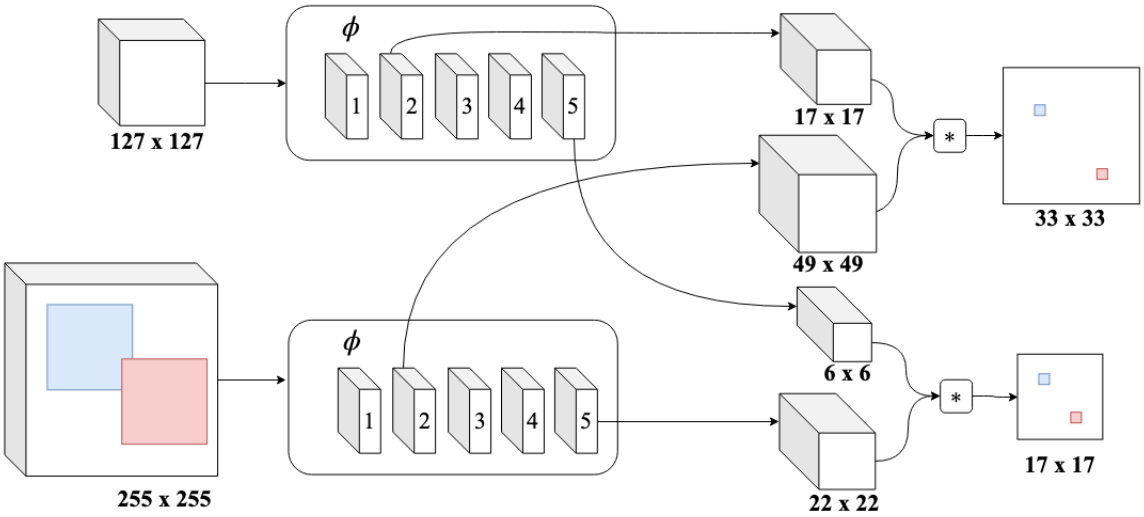


Figure 3.4: Block diagram of Siamese network with fused coarse-fine score maps

3.2.3 Siamese network with modified residual blocks

In SiamFC tracker, the backbone CNN (AlexNet) is shallow in nature, which does not take complete advantage of the capability of modern deep CNNs. In this experiment, we leverage the power of deep convolutional neural networks to enhance tracking robustness and accuracy. We observe that directly replacing the backbones with existing powerful architectures, such as ResNet does not improve the tracking perspective. After analyzing the siameseFC network, there are two inferred reasons that control the tracking performance. 1) Increasing the size of receptive field leads to reduction of feature discrimination and localization precision and 2) the padding factor for convolutions blocks induces a positional bias and redundant features in learning. To address these issues, we can built new residual blocks to eliminate the negative impact of padding, and design new architectures using these blocks with controlled size of receptive field and effective network stride. The designed architectures are lightweight in nature to guarantee real-time tracking speed.

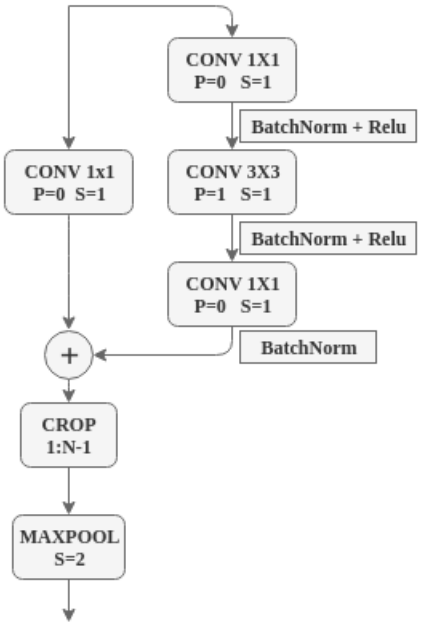


Figure 3.5: Modified residual block

Stage	Modified ResNet-22
conv1	$7\times7, 64$, stride 2
	2×2 MaxPool, stride 2
conv2	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$
conv3	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$ $[1\times1, 128]\times4$
Cross-correlation	
RF	93
OFS	5

Table 3.1: Architecture details of SiamResNet22

3.3 State-of-art: Generative adversarial networks

Generative adversarial networks were introduced by (Goodfellow *et al.* (2014)). GANs are neural networks that generate synthetic data given certain input data. For example, GANs can be taught how to generate images from text. In recent times, GANs have a wide range of useful practical applications with promising results, which include Fake Image generation (Reed *et al.* (2016)), Text-to-image synthesis (Reed *et al.* (2016)), Face aging (Huang *et al.* (2017)), Image-to-image translation, Video synthesis (Vondrick *et al.* (2016)), High-resolution image generation (Ledig *et al.* (2017)) and Image in-painting(Pathak *et al.* (2016)).Generative Adversarial Networks comprises two convolutional neural networks as shown in fig 3.6

Discriminator Network:

The discriminative model operates like a normal binary classifier which is able to classify images into different categories. It determines whether an image is real and from a given dataset or is artificially generated. The discriminator network is a convolutional neural network that acts as a binary classifier.

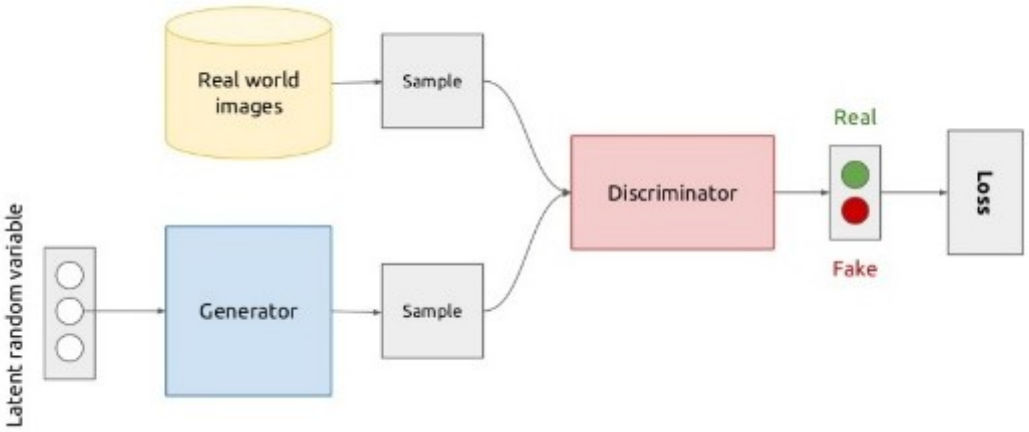


Figure 3.6: Basic block diagram of GAN. [source]

Generator Network:

The discriminative model tries to predict certain classes given certain features. The generative model tries to predict features given classes. This involves determining the probability of a feature given a class. The generator produces new realistic images through a deconvolutional neural network.

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

---

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.
- Sample minibatch of m examples $\{x^{(1)}, \ldots, x^{(m)}\}$ from data generating distribution $p_{\text{data}}(x)$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(x^{(i)}\right) + \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right) \right].$$

end for

- Sample mini batch of m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

3.3.1 Theoretical Foundations

In a min-max concept of game theory, a GAN has two players which are a generator and a discriminator. A generator generates new instances of an object while the discriminator determines whether the new instance belongs to the original dataset. During the training process, weights and biases are adjusted through back propagation until the discriminator learns to distinguish real images from fake images. The generator gets feedback from the discriminator and uses it to produce images that are more realistic. The step-by-step approach of GAN training has been explained in the algorithmic chart 1

3.3.2 Different types of GANs

Context-Conditional GAN

In CCGAN Denton *et al.* (2016), they introduce a simple semi-supervised learning approach for images based on in-painting using an adversarial loss. Images with random patches removed are presented to a generator whose task is in-painting the hole, based on the surrounding pixels. The in-painted images are then presented to a discriminator network that judges if they are real (unaltered training images) or not. This task acts as a regularizer for standard supervised training of the discriminator. Using our approach we are able to directly train large VGG-style networks in a semi-supervised fashion. We evaluate on STL-10 and PASCAL datasets, where our approach obtains performance comparable or superior to existing methods.

Figure 3.7: In-painting using CCGAN.

Super-Resolution GAN

Using Deep CNNs, the accuracy and speed of image super-resolution is tackled: Problem of recovering the finer texture details on super-resolving at large up-scaling factors still exists. The optimization-based methods are principally driven by the choice of the objective function. Recent works using the MSE reconstruction have high peak SNR ratios by lacking high-frequency details. SRGAN Ledig *et al.* (2017) is the first framework capable of inferring photo-realistic natural images for 4x up-scaling factors. They propose a perceptual loss function which consists of an adversarial loss and a content loss. The adversarial loss pushes our solution to the natural image manifold using a discriminator network, trained to differentiate the super-resolved images and original photo-realistic images. In addition, they use perceptual similarity as content loss rather than pixel-wise.

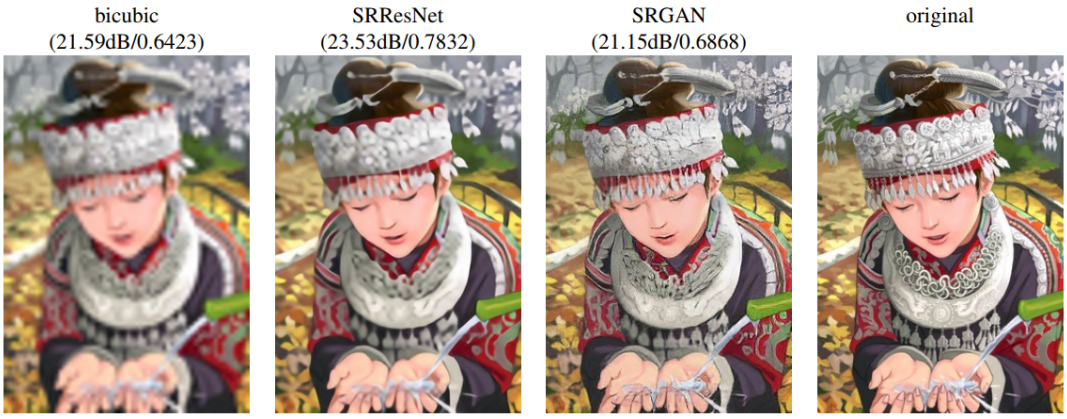


Figure 3.8: Photo-Realistic Image Super-Resolution using SRGAN.

3.4 Proposed approach: ALTO

In (Song *et al.* (2018)), they proposed an adversarial framework that generates random masks to maintain the robust feature description of the target over temporal span in a tracking framework. Unlike the above approach, the key concept of our framework is the integration of similarity-learning CNN into the GAN for leveraging the benefits such as balanced accuracy, high speed and high-end generative power. We propose that adversarial learning can improve the precise target localization through the feedback from patch based discriminator, classifying the predicted target patch cropped using the response

map and the actual ground-truth patch. To make training procedure converge fast, we incorporate the additional constraint such as distance regression between the predicted and the actual target co-ordinates. Unlike the conventional GANs, our generator is a similarity learning CNN which predicts the trajectory of the target motion by minimizing the error between the predicted and the actual trajectory through the adversarial learning.

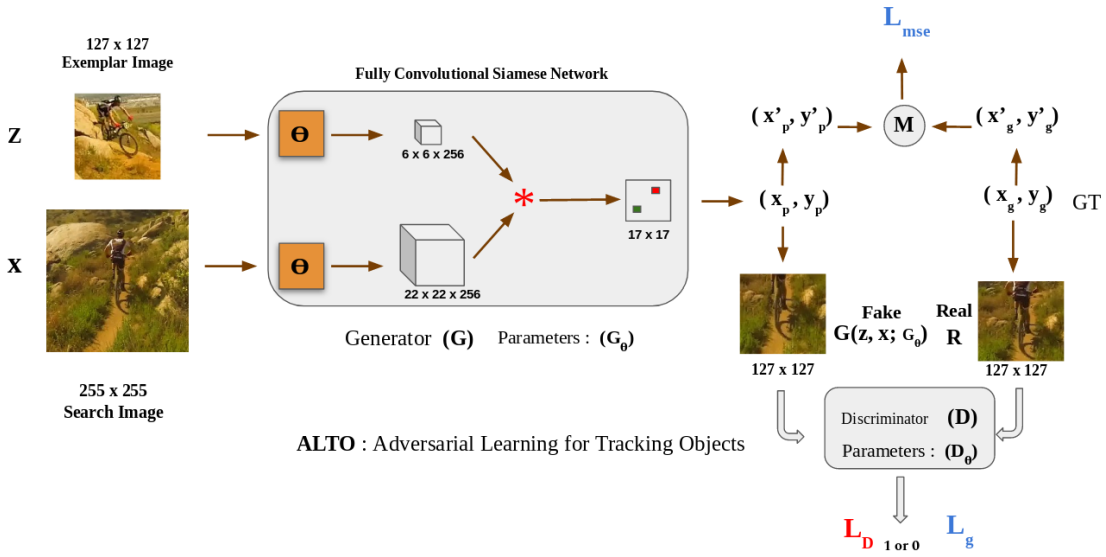


Figure 3.9: ALTO : Adversarial Learning for Tracking Objects. The response map is a scalar valued indicating the similarity score for the positive samples (green) and negative samples (red).

3.4.1 Training procedure

We adapt fully convolutional siamese network (Bertinetto *et al.* (2016)) as a generative model **G** for response maps by performing cross-correlation between the embeddings, obtained from a template image z and search image x as shown in 3.9

$$G(z,x) = \theta(z) * \theta(x) + b\,1, \tag{3.9}$$

Using the generated response map, the location with high correlation score is determined. Then, we crop a 127 pixels square patch from the search image centered with predicted location. We then term the cropped patch as a fake sample and feed it to the discriminator **D**, while the template patch is termed as the real sample. We employ the optimization function as binary cross entropy L, as given by eq.(3.10) to model the

adversarial loss, where y and $\bar{y}$ denote true and predicted probabilities, respectively.

$$L = -(y \log \bar{y} + (1-y) \log (1-\bar{y})) \tag{3.10}$$

During the training, the generator attempts in modelling its output such that the discriminator classifies the predicted patch to be a real sample i.e. converging to the target template. Thus the predicted probability $\bar{y} = \mathbf{D}\Big(\mathbf{G}(z, x \mid \mathbf{G}_\theta) \mid \mathbf{D}_\theta\Big)$ and true probability $y = 1$ results in formulating the generator loss L_g, as given by Eq. (3.11).

$$L_g = -\log\left(\mathbf{D}\Big(\mathbf{G}(z, x \mid \mathbf{G}_\theta) \mid \mathbf{D}_\theta\Big)\right) \tag{3.11}$$

To provide a better and faster convergence in learning procedure of the adversarial framework, we incorporate an additional optimization term in the Generator loss, that quantitatively improve the proximity of the predicted location by the Generator to the actual location. Specifically, we add a regression constraint to the Generator loss, that helps in minimizing the error between the predicted target location in the response map (x_p, y_p) and the actual ground-truth target location (x_t, y_t), as depicted in Fig. **??**. Therefore, the overall loss for the generator is given by Eq. (3.12). Finally the parameters $\mathbf{G}_\theta$ of the generator are learned by minimizing the cost function Eq.(3.12) with respect to $\mathbf{G}_\theta$ through stochastic gradient descent (SGD).

$$L_G = L_g + L_{mse} \tag{3.12}$$

$$\arg\min_{\mathbf{G}_\theta} \mathop{\mathbb{E}}_{(z,x)} L_G \tag{3.13}$$

The training process of the generator and the discriminator is carried out together to achieve the common equilibrium. The discriminator loss contains two components real loss L_{real} and fake loss L_{fake} . L_{real} can be modelled as L with $y = 1$ and $\bar{y} = \mathbf{D}(R \mid \mathbf{D}_\theta)$, where R being the true target image. Similarly L_{fake} can be modelled as L with $y = 0$ and $\bar{y} = \mathbf{D}\Big(\mathbf{G}(z, x \mid \mathbf{G}_\theta) \mid \mathbf{D}_\theta\Big)$ where $\mathbf{G}(z, x \mid \mathbf{G}_\theta)$ is the generator predicted patch in search

region. The total loss of the discriminator L_D is formed as summation of real loss and fake loss.

$$L_{real} = -\log\Big(\mathbf{D}(R \mid \mathbf{D}_\theta)\Big) \tag{3.14}$$

$$L_{fake} = -\log\Big(1 - \mathbf{D}\Big(\mathbf{G}(z,x \mid \mathbf{G}_\theta) \mid \mathbf{D}_\theta\Big)\Big) \tag{3.15}$$

Finally, the model parameters $\mathbf{D}_\theta$ of the discriminator network are learned by minimizing Eq. (3.15) with respect to $\mathbf{D}_\theta$. This kind of discriminative learning encourages the generator to predict image patch that is more closer to the target and thereby improve tracking accuracy.

Stage	**Generator**	**Discriminator**
Convolution1	11×11, 96, stride 2 [*]	3×3, 64, stride 2 [**]
3×3 MaxPool, stride 2		
Convolution2	5×5, 256, stride 1 [*]	3×3, 128, stride 2 [**]
3×3 MaxPool, stride 2		
Convolution3	3×3, 384, stride 1 [*]	3×3, 256, stride 1 [**]
Convolution4	3×3, 384, stride 1 [*]	3×3, 512, stride 1 [**]
Convolution5	3×3, 256, stride 1	3×3, 1, stride 1
	Cross-correlation	
Receptive field	127	-
Output feature size	6	1

[$*$]- *Batchnorm + Relu* | [$**$]- *Batchnorm + Leaky Relu*

Table 3.2: Architecture Details of ALTO

3.4.2 Tracking procedure

During the phase of online tracking, we only retain the generator and a search image patch centered around previous target location is used for prediction of target's location in the current frame. The location of the high score in the response map relative to the centre of the score map, multiplied by the stride of the network gives the displacement

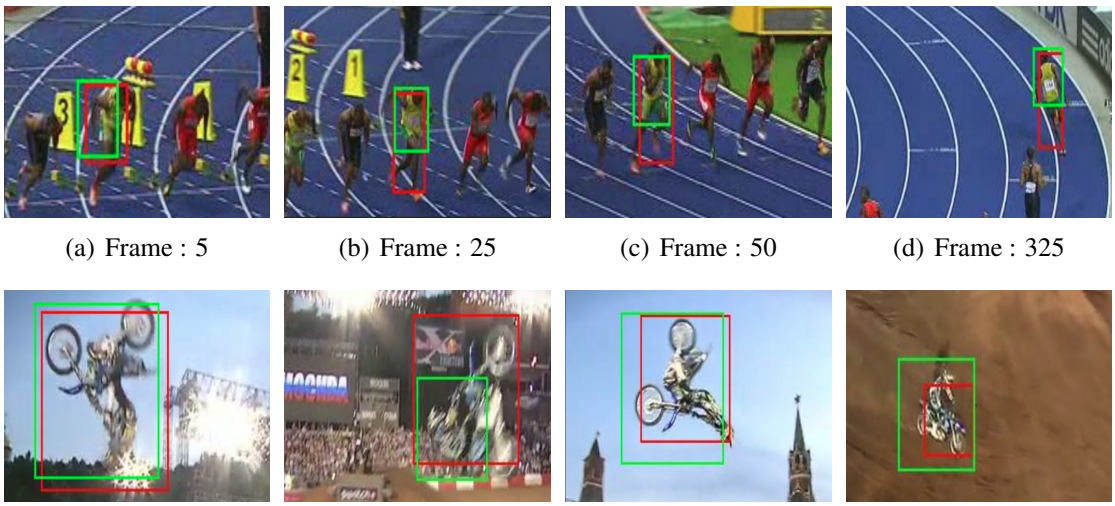
of the target in search image. Then, it performs a naive multi-scale search for estimating target size to update the previous size, which is highly sensitive to target location.

We carry out the training for 50 epochs using one NVIDIA GTX-1080Ti on ILSVRC 2015 dataset (Russakovsky *et al.* (2015)) consisting of 4500 videos with almost 1.5 million annotated frames of 30 different categories. For both networks, we employ binary cross entropy loss with Stochastic Gradient Descent (SGD) for updating the network parameters. In case of generator, we employ the additional constraint mean square error between the predicted location and the ground truth. We use the optimizer with the momentum 0.9 and a weight decay 0.0005 for convolution layer parameters in both generator and discriminator. We evaluate the proposed method on popular object tracking benchmarks like OTB (Wu *et al.* (2015)), Temple128 (Liang *et al.* (2015)), DTB (Li and Yeung (2017)), and VOT2016 (Kristan *et al.* (2016)). Our results show that the proposed approach ALTO outperforms the baseline approach on these challenging benchmarks. We also compare ALTO with popular tracking methods KCF (Henriques *et al.* (2015)), HCF (Ma *et al.* (2015)), SRDCF (Danelljan *et al.* (2015*b*)), CCOT (Danelljan *et al.* (2016)), VITAL (Song *et al.* (2018)). We evaluate the performance of ALTO using VOT (Kristan *et al.* (2016)) and GOT (Huang *et al.* (2018)) tool kits.

CHAPTER 4

QUALITATIVE RESULTS AND DISCUSSIONS

We show some qualitative results of our tracker on standard benchmark sequences. The results of the tracker on two sequences from VOT2016 dataset is shown in the Figure 4.1. The first sequence (bolt1) contain frequent changes in target pose leading to large changes in the target state, and is well suited to evaluate our approach. The second sequence (motocross1sequence) contain the geometric and photo-metric variations such as rotations and illumination changes. For comparison, we included the output of the state-of-the-art Siamese FC tracker. These approaches employ multi-scale search algorithm for determining the target scale i.e width and height of the bounding box. The results show that our adversarial learning approach improves the target localization indeed improving the target scale estimation.



(a) Frame : 5 (b) Frame : 25 (c) Frame : 50 (d) Frame : 325

(e) Frame : 5 (f) Frame : 75 (g) Frame : 100 (h) Frame : 155

Figure 4.1: Comparison of our approach **ALTO** (red) and baseline approach **SiamFC** (green) on bolt1 sequence and motocross1 sequence from VOT 2016. Note that results are best viewed in color.

We compare the original shallow SiamFC- AlexNet with deep SiamResNet22 with OTB15, OTB15, VOT16 and VOT17 benchmark datasets to prove that deep state-of-art architectures can improve tracking accuracy and robustness with necessary modifications in impact parameters.

Model	OTB13	OTB15	VOT16	VOT17
AlexNet	**0.608**	**0.579**	**0.235**	**0.188**
ResNet22	**0.663**	**0.644**	**0.303**	**0.234**

Table 4.1: Comparison between shallow AlexNet and deep ResNet22 siamese architectures.

Our experiments show that the proposed approach outperforms the baseline approach and performs better than many state-of-the-art methods on the VOT2016 benchmark. Though the proposed adversarial learning based tracking framework is demonstrated with Siamese network, mainly due to its simplicity, the key idea can be extended to other versions of Siamese trackers for the generalization.

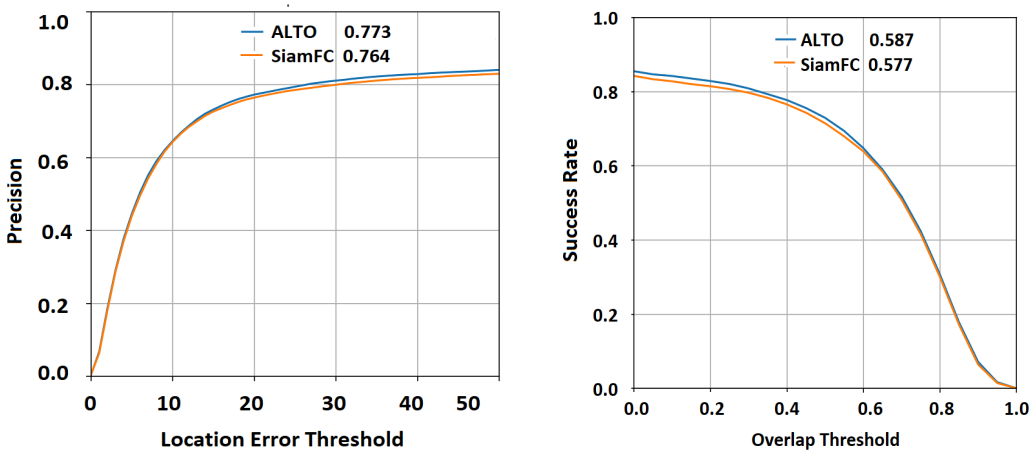
Tracker	Accuracy	Robustness	EAO
ALTO	**0.5567**	26.83	*0.2744*
CCOT	0.5351	**14.00**	**0.3310**
HCF	0.4354	*20.50*	0.2206
KCF	0.5065	31.66	0.1924
SRDCF	0.5318	24.33	0.2471
SiamFC	*0.5382*	21.50	0.2606
VITAL	***0.5438***	***16.50***	***0.3228***

Table 4.2: State-of-the-art comparison on VOT2016 Dataset in terms of accuracy and robustness. Our approach significantly outperforms baseline and many state-of-art trackers by achieving a relative gain in given metrics.

Tag	ALTO	SiamFC	VITAL
	Accuracy	**Accuracy**	**Accuracy**
Camera motion	**0.5430**	***0.5378***	0.5363
Empty	**0.6015**	***0.5733***	0.5732
Illumination	**0.6764**	***0.6738***	0.5750
Motion Change	***0.5262***	0.5039	**0.5383**
Occlusion	***0.4800***	0.4356	**0.5154**
Size Change	***0.5132***	0.5046	**0.5244**
Mean Accuracy	**0.5567**	0.5382	***0.5438***

Table 4.3: State-of-the-art comparison under different tags on VOT2016 dataset between ALTO, SiamFC (*baseline*) and VITAL in terms of the accuracy.

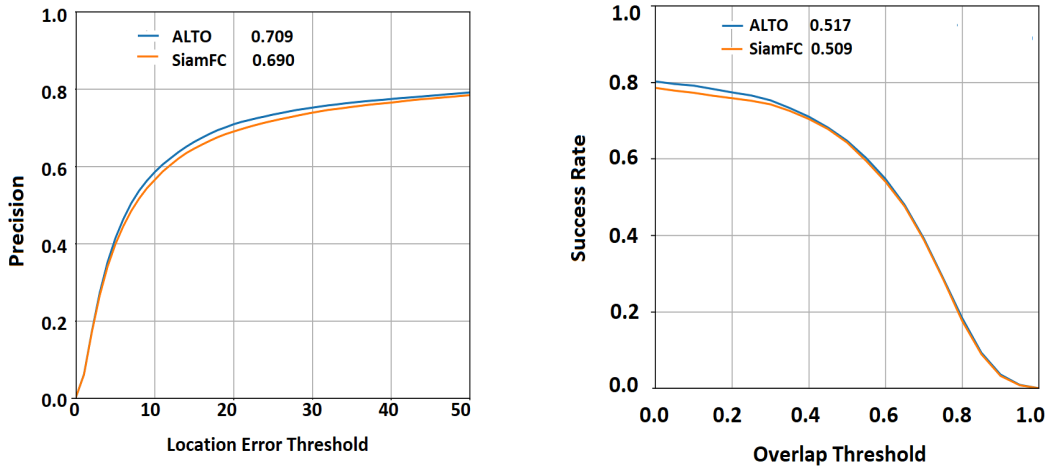
We compare our proposed approach with SiamFC. Figure 4.2 displays the success plot and precision plot over all OTB100 benchmark challenging videos. Our tracker, employing adversarial learning for target localization, significantly outperforms SiamFC by achieving AUC of significant percent.



(a) (b)

Figure 4.2: Comparison between ALTO and SiamFC in terms of Precision (a) and Success (b) on OTB100 benchmark dataset.

We compare our proposed approach with SiamFC. Figure 4.3 displays the success plot and precision plot over Temple128 dataset. Our tracker, employing adversarial learning for target localization, significantly outperforms SiamFC by achieving AUC of significant percent.



(a) (b)

Figure 4.3: Comparison between ALTO and SiamFC in terms of Precision (a) and Success (b) on Temple128 dataset.

CHAPTER 5

CONCLUSION

In this thesis, the problem of accurate localization in tracking was studied and addressed. The proposed framework ALTO enables better location prediction of the target in similarity-learning trackers through incorporation of the adversarial learning, through correction of the predictions made by the baseline tracker. This approach was shown to provide the best results on the well known challenging tracking datasets, outperforming other state-of-the-art trackers, thus demonstrating the impact of the proposed approach. Though the proposed framework is demonstrated with Siamese network, mainly due to its simplicity, the key concept of the idea can be extended to other versions of Siamese trackers.

Hence, we conclude that this kind of adversarial framework for tracking has paramount importance and holds unprecedented scopes for further improvements in the field of object tracking. There is a necessity for extensive baseline experiments to investigate and understand the impact of the various design choices on the proposed methodology.

REFERENCES

1. **B. C. Arvind**, **S. K. Nagaraj**, **C. S. Seelamantula**, and **S. S. Gorthi**, Active-disc-based kalman filter technique for tracking of blood cells in microfluidic channels. *In 2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016.

2. **L. Bertinetto**, **J. Valmadre**, **J. F. Henriques**, **A. Vedaldi**, and **P. H. Torr**, Fully-convolutional siamese networks for object tracking. *In European conference on computer vision*. Springer, 2016.

3. **G. Bhat**, **J. Johnander**, **M. Danelljan**, **F. Shahbaz Khan**, and **M. Felsberg**, Unveiling the power of deep tracking. *In Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.

4. **D. S. Bolme**, **J. R. Beveridge**, **B. A. Draper**, and **Y. M. Lui**, Visual object tracking using adaptive correlation filters. *In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010.

5. **C. Chen**, **A. Seff**, **A. Kornhauser**, and **J. Xiao**, Deepdriving: Learning affordance for direct perception in autonomous driving. *In Proceedings of the IEEE International Conference on Computer Vision*. 2015.

6. **N. Dalal** and **B. Triggs**, Histograms of oriented gradients for human detection. *In international Conference on computer vision & Pattern Recognition (CVPR'05)*, volume 1. IEEE Computer Society, 2005.

7. **M. Danelljan**, **G. Bhat**, **F. Shahbaz Khan**, and **M. Felsberg**, Eco: efficient convolution operators for tracking. *In Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

8. **M. Danelljan**, **G. Hager**, **F. Shahbaz Khan**, and **M. Felsberg**, Convolutional features for correlation filter based visual tracking. *In Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2015*a*.

9. **M. Danelljan**, **G. Hager**, **F. Shahbaz Khan**, and **M. Felsberg**, Learning spatially regularized correlation filters for visual tracking. *In Proceedings of the IEEE international conference on computer vision*. 2015*b*.

10. **M. Danelljan**, **A. Robinson**, **F. S. Khan**, and **M. Felsberg**, Beyond correlation filters: Learning continuous convolution operators for visual tracking. *In European Conference on Computer Vision*. Springer, 2016.

11. **E. Denton**, **S. Gross**, and **R. Fergus** (2016). Semi-supervised learning with context-conditional generative adversarial networks. *arXiv preprint arXiv:1611.06430*.

12. **I. Goodfellow**, **J. Pouget-Abadie**, **M. Mirza**, **B. Xu**, **D. Warde-Farley**, **S. Ozair**, **A. Courville**, and **Y. Bengio**, Generative adversarial nets. *In Advances in neural information processing systems*. 2014.

13. **E. Gundogdu** and **A. A. Alatan** (2018). Good features to correlate for visual tracking. *IEEE Transactions on Image Processing*, **27**(5), 2526–2540.

14. **D. Held**, **S. Thrun**, and **S. Savarese**, Learning to track at 100 fps with deep regression networks. *In European Conference on Computer Vision*. Springer, 2016.

15. **J. F. Henriques**, **R. Caseiro**, **P. Martins**, and **J. Batista** (2015). High-speed tracking with kernelized correlation filters. *IEEE transactions on pattern analysis and machine intelligence*, **37**(3), 583–596.

16. **C.-M. Huang** and **L.-C. Fu** (2011). Multitarget visual tracking based effective surveillance with cooperation of multiple active cameras. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, **41**(1), 234–247.

17. **L. Huang**, **X. Zhao**, and **K. Huang** (2018). Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *arXiv preprint arXiv:1810.11981*.

18. **R. Huang**, **S. Zhang**, **T. Li**, and **R. He**, Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. *In Proceedings of the IEEE International Conference on Computer Vision*. 2017.

19. **H. Kiani Galoogahi**, **A. Fagg**, and **S. Lucey**, Learning background-aware correlation filters for visual tracking. *In Proceedings of the IEEE International Conference on Computer Vision*. 2017.

20. **G. Koch**, **R. Zemel**, and **R. Salakhutdinov**, Siamese neural networks for one-shot image recognition. *In ICML deep learning workshop*, volume 2. 2015.

21. **M. Kristan**, **J. Matas**, **A. Leonardis**, **T. Vojir**, **R. Pflugfelder**, **G. Fernandez**, **G. Nebehay**, **F. Porikli**, and **L. Čehovin** (2016). A novel performance evaluation methodology for single-target trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **38**(11), 2137–2155. ISSN 0162-8828.

22. **C. Ledig**, **L. Theis**, **F. Huszár**, **J. Caballero**, **A. Cunningham**, **A. Acosta**, **A. Aitken**, **A. Tejani**, **J. Totz**, **Z. Wang**, *et al.*, Photo-realistic single image super-resolution using a generative adversarial network. *In Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

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

24. **S. Li** and **D.-Y. Yeung**, Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models. *In AAAI*. 2017.

25. **P. Liang**, **E. Blasch**, and **H. Ling** (2015). Encoding color information for visual tracking: Algorithms and benchmark. *IEEE Transactions on Image Processing*, **24**(12), 5630–5644.

26. **A. Lukezic**, **T. Vojir**, **L. Cehovin Zajc**, **J. Matas**, and **M. Kristan**, Discriminative correlation filter with channel and spatial reliability. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

27. **C. Ma**, **J.-B. Huang**, **X. Yang**, and **M.-H. Yang**, Hierarchical convolutional features for visual tracking. *In Proceedings of the IEEE international conference on computer vision*. 2015.

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

29. **H. S. Parekh**, **D. G. Thakore**, and **U. K. Jaliya** (2014). A survey on object detection and tracking methods. *International Journal of Innovative Research in Computer and Communication Engineering*, **2**(2), 2970–2979.

30. **O. M. Parkhi**, **A. Vedaldi**, **A. Zisserman**, *et al.*, Deep face recognition. *In bmvc*, volume 1. 2015.

31. **D. Pathak**, **P. Krähenbühl**, **J. Donahue**, **T. Darrell**, and **A. Efros**, Context encoders: Feature learning by inpainting. *In Computer Vision and Pattern Recognition (CVPR)*. 2016.

32. **S. Reed**, **Z. Akata**, **X. Yan**, **L. Logeswaran**, **B. Schiele**, and **H. Lee** (2016). Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*.

33. **S. Ren**, **K. He**, **R. Girshick**, and **J. Sun**, Faster r-cnn: Towards real-time object detection with region proposal networks. *In Advances in neural information processing systems*. 2015.

34. **O. Russakovsky**, **J. Deng**, **H. Su**, **J. Krause**, **S. Satheesh**, **S. Ma**, **Z. Huang**, **A. Karpathy**, **A. Khosla**, **M. Bernstein**, **A. C. Berg**, and **L. Fei-Fei** (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, **115**(3), 211–252.

35. **A. Sharif Razavian**, **H. Azizpour**, **J. Sullivan**, and **S. Carlsson**, Cnn features off-the-shelf: an astounding baseline for recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2014.

36. **Y. Song**, **C. Ma**, **X. Wu**, **L. Gong**, **L. Bao**, **W. Zuo**, **C. Shen**, **R. W. Lau**, and **M.-H. Yang**, Vital: Visual tracking via adversarial learning. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

37. **C. Sun**, **D. Wang**, **H. Lu**, and **M.-H. Yang**, Correlation tracking via joint discrimination and reliability learning. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

38. **Y. Taigman**, **M. Yang**, **M. Ranzato**, and **L. Wolf**, Deepface: Closing the gap to human-level performance in face verification. *In Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.

39. **R. Tao**, **E. Gavves**, and **A. W. Smeulders**, Siamese instance search for tracking. *In Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

40. **J. Valmadre**, **L. Bertinetto**, **J. Henriques**, **A. Vedaldi**, and **P. H. Torr**, End-to-end representation learning for correlation filter based tracking. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

41. **J. Van De Weijer**, **C. Schmid**, **J. Verbeek**, and **D. Larlus** (2009). Learning color names for real-world applications. *IEEE Transactions on Image Processing*, **18**(7), 1512–1523.

42. **C. Vondrick**, **H. Pirsiavash**, and **A. Torralba**, Generating videos with scene dynamics. *In Advances In Neural Information Processing Systems*. 2016.

43. **Q. Wang**, **Z. Teng**, **J. Xing**, **J. Gao**, **W. Hu**, and **S. Maybank**, Learning attentions: residual attentional siamese network for high performance online visual tracking. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

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

45. **S. Zagoruyko** and **N. Komodakis**, Learning to compare image patches via convolutional neural networks. *In Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

46. **Z. Zhu**, **Q. Wang**, **B. Li**, **W. Wu**, **J. Yan**, and **W. Hu**, Distractor-aware siamese networks for visual object tracking. *In Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.

Publications based on this thesis

1. **Naveen Paluru*, Bala Suraj*, Litu Rout** and **Rama Krishna Gorthi**, "ALTO: Adversarial Learning for Tracking Objects", *Pattern Recognition Letters, 2019* (Under Review) (* denotes Equal contribution)