12-15-2018

UAS-based Object Tracking via Deep Learning

Marc Dinh
mind.char@gmail.com

UAS-BASED OBJECT TRACKING VIA DEEP LEARNING

By

Marc Dinh

Bachelor of Science (B.Sc.)
University of Paris-Est, Marne-la-Vallee
2013

A thesis submitted in partial fulfillment
of the requirements for the

Master of Science in Computer Science

Department of Computer Science
Howard R. Hughes College of Engineering
The Graduate College

University of Nevada, Las Vegas
December 2018

UNLV GRADUATE COLLEGE

Thesis Approval

The Graduate College
The University of Nevada, Las Vegas

November 16, 2018

This thesis prepared by

Marc Dinh

entitled

UAS-Based Object Tracking via Deep Learning

is approved in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science
Department of Computer Science

Yoohwan Kim, Ph.D.
Examination Committee Chair

Fatma Nasoz, Ph.D.
Examination Committee Member

Justin Zhan, Ph.D.
Examination Committee Member

Brendan Morris, Ph.D.
Graduate College Faculty Representative

Kathryn Hausbeck Korgan, Ph.D.
Graduate College Interim Dean

Abstract

Tracking is the task of identifying an object of interest and detect its position over time, and has numerous applications like surveillance, security and traffic control. In present times, unmanned aerial vehicles (UAV) have been more and more common which provides us with a new and less explored domain, with an ideal vantage point for surveillance and monitoring applications.. Aerial tracking is a particularly challenging task as it introduces new environmental variables such as rapid motion in 3D space. We propose a new deep learned tracker architecture catered to aerial tracking.

First, a study of six state-of-the-art deep learned trackers has been performed using the Visual Object Tracking benchmark. This study determined the weaknesses of said trackers in front of a long-term aerial tracking task. Mainly, severe motion, target disappearance and high degree of appearance change were the principal causes for drift or loss of track.

Siamese correlation filter based tracker to perform identification and target matching across subsequent frames. In addition, a multi-scale object detector has been implemented to improve identification accuracy and template update. The object detector goal is to output a score map that will validate or penalize the tracker's correlation map, and improve robustness against drift, scale change and occlusion.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Visual object tracking is the task of recognizing an object and match its position within subsequent frames. Visual object tracking is a challenging task in computer vision and has many applications in the industry. From security and surveillance to traffic control, photography and video, tracking has gathered a great research interest. In the present time, correlation based tracker are the state of the art technology. [CLK15]

Unmanned aerial vehicles (UAV) have been increasing in the airspace. Initially developed for military purposes, they provide an ideal vantage point for surveillance, security and traffic control applications[SSAF$^+$18] [IMJ$^+$15]. However, current research do not test for long term tracking or UAV view-points.[MSG16] An aerial moving camera introduces a new set of variables that are yet to be addressed by a visual system.

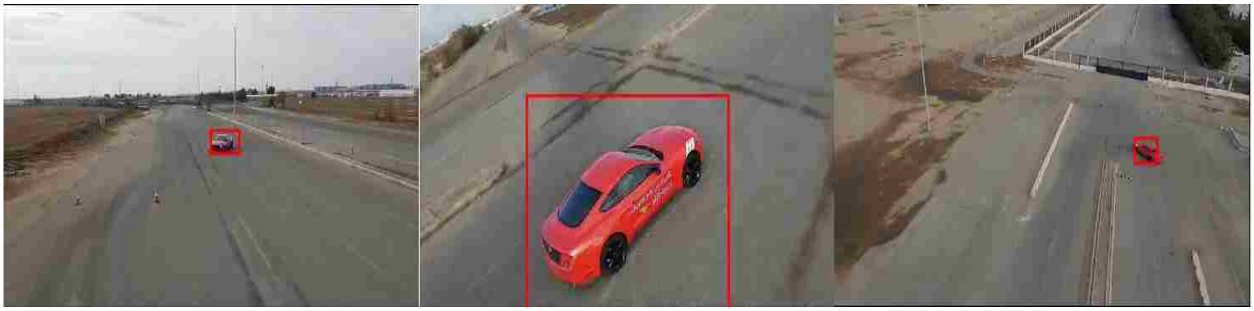


Figure 1.1: .Tracking on challenging data: object deformation and scale change in dataset car16

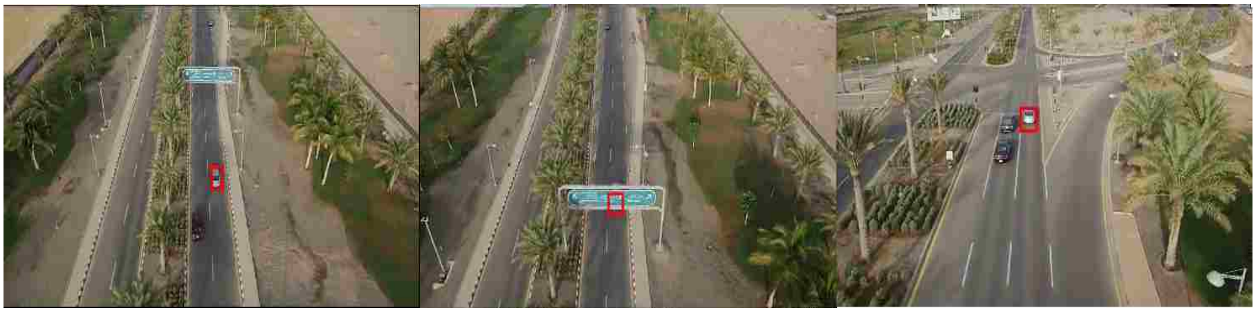


Figure 1.2: .Tracking on challenging data: object disappearance and clutter in dataset car3

Traditionally a tracker outputs a bounding box around an object of interest. Previous works are focused on short-term sequences (tens of seconds) without object disappearance and complex transformations[KLM+17]. Specifically, long term tracking is focused on sequences that are longer than a minute with frequent object disappearance and appearance change.[VBH+18] In this work, we propose an original and intelligent vision system that identify and track a vehicle from an UAV in high speed.

Deep learning has accomplished great advancements in computer vision. We propose a tracker that fully takes advantage of the power and expressiveness of deep learning.

1.2 Goals

Tracking is a known an popular challenge in the computer vision research area. The goals of the proposed model is to identify and track fast moving object in real-time while the UAV is airborne.

Tracking has to deal complex appearance change, such as out-of-plane rotation, non-rigid deformation, camera perspective, motion blur, illumination changes, occlusions and clutter [WLY13].

Due to relative motion of the target to the UAV in 3D space, object appearance can dramatically change from one frame to the other. Many models use an internal model that is updated every frame. Accuracy of the model decreases over time with the accumulation of error, resulting in a drift. Moreover, motion models are harder to learn in order to separate target motion from UAV motion. As a result, the proposed system should have a better internal template update.

Occlusion by ground obstacles such as bridges, buildings or signs is a second hurdle to the success of a track. The system must re-detect the object in the next frame after occlusion. Re detection after occlusion is required.

Because of the newness of UAS-based tracking, we have to conduct our own benchmark and own evaluation, catered to UAS-based tracking.

1.3 Contributions

In this work, we integrate the techniques for Discriminative Correlation Filter tracking using a fully convolutional siamese network with a very fast deep learned object detector, and develop a long-term, error-free vehicle tracker for aerial sequences. When compared to the competition, increased precision and IOU has been observed. The proposed approach can detect and track a vehicle of interest in a long-term sequence, even with object deformation, scale variation, object disappearance and rapid motion. The main contributions are three-fold:

- The new model is able to generate an adaptive scale bounding box in real-time around the vehicle being tracked, as the target changes dimensions and appearance.
- Our model, using multi-scale detection can handle object disappearance and long-term tracking
- The proposed system runs in real-time with an FPS greater than 10.

1.4 Outline

In chapter 2, Past research on state-of-the-art deep learned trackers and object detectors is presented

In chapter 3, the fully-convolutional siamese networks architecture will be detailed.

In chapter 4, real time object detection will be thoroughly studied. End to end training will also be explained.

In chapter 5, our novel system for robust aerial tracking will be presented. A forward pass will be detailed for better understanding and detailed implementation is discussed.

Chapter 6 explains the experiments, methodology, learning process, observations and results. A study of the performance of our intelligent system, along with 6 other competing state-of-the-art trackers is performed in 7 challenging sequences. Quantitative and qualitative results are also shown.

Finally, chapter 7 reviews the advantages and shortcomings of our tracker. Future prospects and applications along with further improvements are also presented.

Chapter 2

Literature Review

Traditionally, object tracking is following a Tracking-by-detection paradigm. In other words, tracking is divided into two steps.

First, object detection, where the the target is identified. We assess its location, dimensions and appearance. Then, we try to match frame by frame it's appearance by outputting a bounding box surrounding the object of interest. The tracking is successful when the bounding box exists at every frame of the sequence, and the object is perfectly within the prediction.

2.1 Object Detection

Object detection is equivalent to the separation of an object of interest from the background in an image.[ST17] It has to take into consideration its position and dimension in order to output a accurate prediction. In recent years, the rise of deep learning has rendered obsolete traditional object detection. In the past object detection would be based on mathematical models, describing low level features such as histogram of gradients, edge detection etc. [ST17] for object detection.

The deep learned approach takes advantage of the expressiveness and robustness of feature representation with convolutional neural networks (CNN).

Deep learned object detection are broadly classified as two categories [ZZtXW18].

In a region based approach,[GDDM14] [Gir15] features are extracted at different scales resulting in regions of interest with a finer grain at each iteration. Then detection is performed on the best region of interest.

On the other hand , one-shot detection, where the image is processed in one stage, such as [RDGF15] [RF16] [LAE+16].

2.2 Object Tracking

Because of their impressive high-speed, correlation filters have attracted a great deal of interests in object tracking. In 2017, 49 percent of the trackers submitted to VOT applied a discriminitative correlation filter approach. [KLM+17]

Discrimintative Correlation Trackers are based on the same approach. Detection and matching is done from frame to frame by computing a correlation between an internal representation of the target and a candidate patch.

CRSDCF [LVZ+18] is using low level features, Historgram Of Gradients and color-names for template embedding, ensuring spatial reliability. Thus improving the discriminative correlation response.

Other trackers rely on a deep neural network method to perform template embedding, For instance, Gundogdu et al. [GA17] have proposed a standard CNN to fine-tuned for correlation-based target localization filter, which works by trying to find the best features that maximizes the discriminative correlation response. It uses an optimized backpropagation algorithm to perform online learning on the target features. Thus improving correlation response and robustness in case of appearance change.

Bertinetto et al, [BVH+16] have proposed a model with an original fully-convolutional Siamese network, trained offline on the ILSVRC15 dataset, to tackle object location. In practice, the network is using an efficient sliding-window evaluation in order to locate the object in a search area.

the ECO [DBKF16] tracker takes advantage of CNN feature extraction to match target and observation from frame to frame. It aims to simplify convolution computation by introducing three algorithms :

1. a factorized convolution operator, reducing the number of parameters in the model

2. a compact generative model of the training sample distribution, that dramatically lowers memory overhead and time complexity, while improving the resulting distribution

3. a conservative model update strategy with improved robustness and reduced complexity.

Two other top performing trackers also adopt the ECO tracker. CFWCR [HFZ$^+$17] exploits the CNN feature extraction to calculate the confidence score. It performs feature extraction in multiple channels, resulting in multiple feature maps. Each feature map extracted is normalized and a weighted convolution from each feature are summed to produce the final response.

Finally, gNet [KLM$^+$17] integrates GoogLeNet CNN for feature extraction with SRDCF for spatial reliability and ECO model. the Spatially Regularized DCF (SRDCF) address boundary effects by penalizing coefficients residing outside the target region.

Chapter 3

Fast Real-Time Object Detection

In this chapter the one-shot object detector You Only Look Once (YOLO) [RDGF15] will be detailed. Traditionally, object detection aims to separate an object of interest from the background. The role of object detection in the tracking task, consists of extracting the features of the object, effectively learning an internal representation. During tracking, this representation is matched frame by frame, generating a full track when combined.

The YOLO model directly predicts bounding box and class probabilities. Previous methods like recurrent CNN used a multi step approach with region of interest. This is slow to compute and harder to train. YOLO is performing localization and classification in one step and a single neural network [RF16].

YOLO was selected because it can run at high speed (¿100 frames per seconds) , with an accuracy equivalent or better than the competition[RF16].

3.1 Architecture Overview

YOLO divides the the image in to an SxS grid, with N proposal bounding boxes and C different classes. For each grid cell, the network generates B bounding boxes. For each proposal box, the network outputs a confidence score, or the probability of an object being detected. That equates to a vector of 5 elements $[x, y, w, h, confidence]$. The (x, y) coordinates are the center of the box, relative to the grid cell location, and (w, h) are the dimension of the box. Those coordinates are normalized respectively to the position of the cell in the grid and to the image size. Meaning

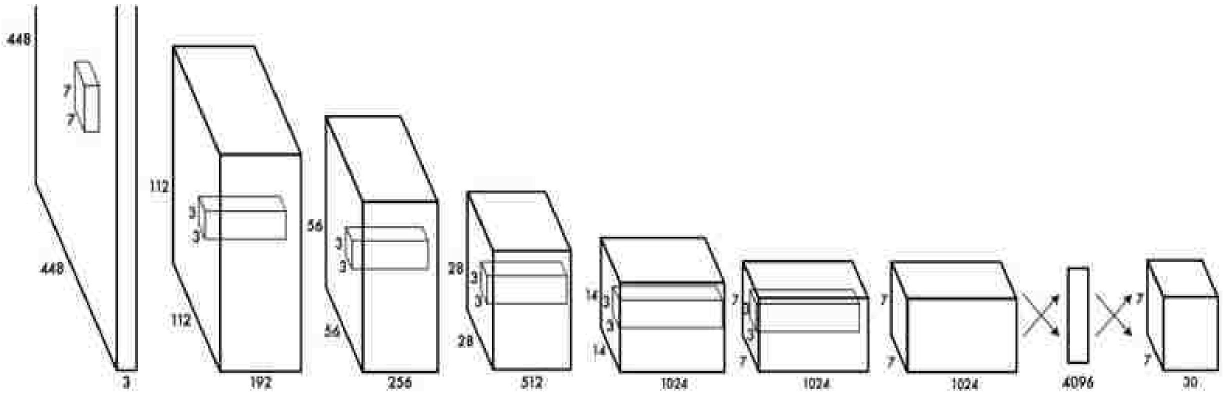


Figure 3.1: Architecture Overview [RDGF15]

$x, y, w, h \in [0, 1]$.

The confidence score reflects the presence or absence of an object in the bounding box, and is defined as :

$$Confidence_{box} = Pr(Obj) * IOU(pred, truth)$$

In addition, C conditional class probabilities are computed. In total, we have $S*S*B$ bounding boxes, with C associated class probabilities, resulting in $SxSx(B*5+C)$

24 convolution layers extract features and reduce the dimensions to a $S*S$ tensor. YOLO performs a linear regression using two fully connected layers to make a $S*S*2$ bounding box predictions.finally, we keep the boxes with a high confidence score.

The class confidence score is computed as:

$$Confidence_{class} = Confidence_{box} * Pr(C|Obj)$$

A linear activation function is used in the last layer and a leaky ReLU is used elsewhere.

Chapter 4

Fully Convolutional Siamese Networks

In the past, object tracking has been addressed by learning an internal representation of the object's appearance and dimension exclusively online, using the video itself as the unique training data[vZLK16]. More recently, several attempts to exploit the expressiveness of deep convolutional networks[ZZtXW18] have been made.

However, a classical approach of artificial neural network (ANN) training is not feasible in real time. Training an ANN requires Stochastic Gradient Descent to optimize a loss function, and is traditionally made offline because of heavy computations. Doing a training online to adapt the weights of the network, would severely compromise the performance of the tracker in real-time.[BVH+16]

The Fully Connected Siamese Network, proposes to address the tracking task as a similarity problem. It compares 2 image patches and solves a cross-correlation. A high correlation means that they contain the same object, thus matching (and tracking) is performed.

SiameseFC combines a siamese architecture for template matching, with a CNN for feature extraction trained on the ILSVRC15 dataset, and a discriminative correlation filter.

The Discriminative Correlation Filter is a method that generates a filter to classify images and their in-plane shifts. It is applicable to a real-time application since it is computed in the frequency domain after a Fourier transform. The computations become trivial, enabling the filter to be re-trained for every frame.

However, previous works that use the Correlation Filter have exploited features that were handcrafted or trained on a different task. By solving the regression problem directly with the features extracted by the CNN, we achieve a closed-loop system. This allows embedding deep features that are strongly represented in the discriminative correlation filter, thus improving the discrimination.

4.1 Architecture Overview

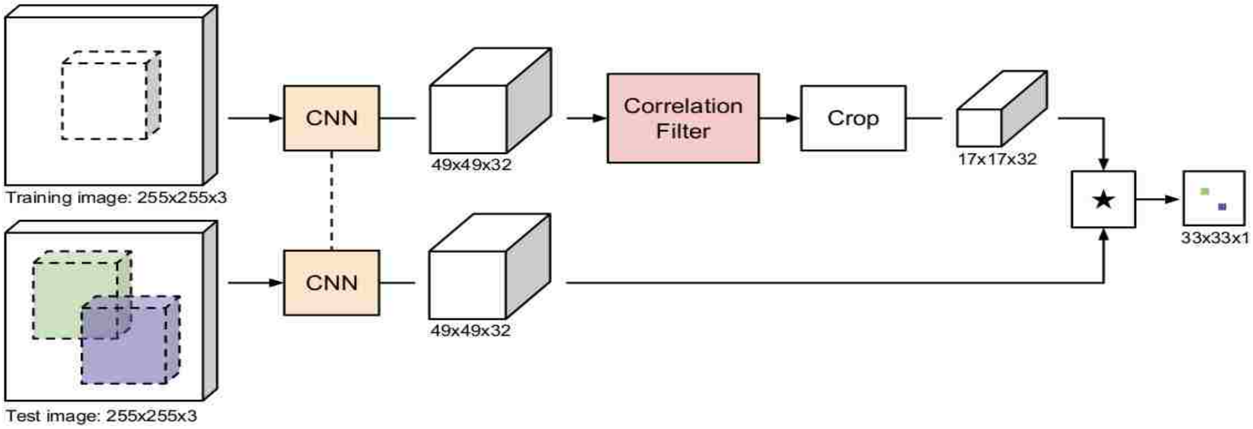


Figure 4.1: Architecture Overview [BVH+16]

4.2 Image and target embedding

Two identical CNN f_ρ are used for feature extraction and dimension reduction. They are trained offline to compare two image patches and determine if they contain the same object. The intuition is that a expressive deep embedding will allow an accurate internal representation thus an accurate matching of objects via similarity.

The Siamese network considers a pair of two images, (x', z'). With x' the representation of the object of interest, and z' the search area. Two feature maps are generated and then cross-correlated. A cross-correlation operation computes the similarity at all translated sub-windows on a dense grid in a single evaluation, rendering the tracker immune to object translation in the plane.

$$g_\rho(x', z') = f_\rho(x') * f_\rho(z') \tag{4.1}$$

The goal is to maximize the value of the response map at the target location. In practice, this is equivalent to searching the pattern x' in the image z'. At run time, the pattern is extracted from the last bounding box, and the test image is a crop centered on the previously estimated position, with a size 4 times larger than the dimension of the object of interest.

As a protection mechanism against appearance change, the template is updated with a rolling average on the current and previous template.

4.3 Correlation Tracking

In order to maximize output response, the correlation filter algorithm is used. A linear classifier is trained on x', to discriminate between the object and its translations. A correlation filter kernel is generated from the feature map extracted from x', by solving a ridge regression in the Fourier domain. let $w(x)$ be the Discriminative Correlation Filter function, that generates a template w, s and b scale factors to compensate for cropping and dimension change. The response map amounts to:

$$g_\rho(x', z') = sw(f_\rho(x')) * f_\rho(z') + b \tag{4.2}$$

4.4 Correlation Filter

Let x be and image with values in $\mathbf{R}^{m*m}$, and w the correlation filter with values in $\mathbf{R}^{m*m}$. We want to minimize:

$$\sum_{u \in U} (\langle x * \delta_{-u}, w \rangle - y[u])^2 = ||w \star x - y||^2 \tag{4.3}$$

$x * \delta_{-u}$. This equation can be understood as the distance between the response map generated by the filter w and the circular shift $x * \delta_{-u}$ of the example image x, and the target signal y.

By introducing regularization parameter λ to prevent over fitting, the problem is to find:

$$\arg\min_{w} \frac{1}{2n} ||w \star x - y||^2 + \frac{\lambda}{2} ||w||^2 \tag{4.4}$$

The optimal template w must satisfy the system of equations :

$$\begin{cases} k &= \frac{1}{n}(x \star x) + \lambda \delta \\ k * \alpha &= \frac{1}{n} y \\ w = \alpha \star x \end{cases} \tag{4.5}$$

where k can be interpreted as the circulant linear kernel matrix, and α is a signal comprised of the Lagrange multipliers of a constrained optimization problem. In the Fourier domain the solution to equation 4.5 is:

$$\begin{cases} \hat{k} = \frac{1}{n}(\hat{x}^* \circ \hat{x}) + \lambda\mathbb{1} \\ \hat{\alpha} = \frac{1}{n}\hat{k}^{-1} \circ \hat{y} \\ \hat{w} = \hat{\alpha}^* \circ \hat{x} \end{cases} \tag{4.6}$$

Where $\hat{x} = Fx$ is the Discrete Fourier Transform of x, x^* is the complex conjugate, $\circ$ is the element-wise multiplication and **1** is a signal of ones. It is critical to note that because we're working in the frequency domain, online learning is now possible, as complex convolutions and correlations are replaced with element-wise multiplications and additions. Thus, the template can be retrained efficiently on every frame, without sacrificing real-time performance.

Chapter 5

Proposed System

In this Chapter, the inner working and implementation of our tracker is explained.Our system integrates an object detector (in pink in 5.3) with a Siamese Correlation tracker (in green 5.3).

First, initialization is done on the first ground truth. Then the Siamese network extracts en exemplar x and a search area z, and the trained DCF estimates the position and dimension of the object. An anomaly is detected when the quality of tracking drops. Low response triggers the re-detection with generation of a score map, followed by updating the template and the new position and dimension of the target object. The new object is then fed to the DCF block for correlation filter training. The proposed re-detection system improves significantly tracking under challenging conditions.

5.1 Algorithm of our proposed tracker

The pseudo code for the tracking algorithm is shown in **Algorithm 1**. Input images are stored in a folder and loaded in memory during evaluation.

Another folder containing the sequence frames with a rectangular bounding box around the target is created during run-time. Four functions make the algorithm.

The tracker is based on a real-time anomaly detection to counteract drift and occlusion. Thus, a real-time anomaly detection algorithm was devised. This policy triggers the activation of the re-detection, re-initialization and adaptive scaling. The workings of this algorithm will be detailed in a later section.

The `main` function initializes the tracker and object detector with the parameters.

The second function `siamese` is performing the correlation operation between the exemplar and the template and returns the score map.

The function `object_detector` performs an object detection on an image and returns several bounding box in a json file, with the coordinates of the center and the object width and height. For each detection, we select the bounding box with the highest confidence score.

The function `template`, trains the correlation filter and outputs a new kernel.

Finally the function `update_position` takes as input the absolute position of the object withing the frame and applies displacement and scale transformation according to the score map.

```
Algorithm 1: Tracker Algorithm
Data: Image frames from the video sequence, $x\_sz$ the size of the search area, n the number
      of frames, t the correlation filter response threshold
Result: Bounding box around target object in each of output image frames of the video
        sequence
initialization;
for $i$ in range($n$):
    initialize scale factors;
    augmentation_map = zeroes(x_sz, x_sz)
    frame, scores, search_area = sess.run([siamese], frame, templatem, pos_x, pos_y)
    /* we calculate the range of correlation values                                  */
    min_score = score.min() max_score = score.max()
    /* we trigger re-initialization if response quality drops                        */
    if $max_score - min_score \leq t$:
        detection = object_detector(search_area)
        for  x, y coordinates in detected bounding box:
            augmentation_map[x][y] = max_score
        target_h = detection.height() target_w = detection.width()
        score_update = (1 -learning_rate) * score + (learning_rate) * augmentation_map
        pos_x, pos_y = update_position(pos_x, pos_y, score_updated, search_size)
        bbox[i,:] = pos_x-target_w/2, pos_y-target_h/2, target_w, target_h
    if $pos\_x \geq frame.width()orpos\_y \geq frame.height()ormax\_score \leq 0$:
        detection = object_detector(frame)
        $target_h = detection.height()$
        $target_w = detection.width()$
        pos_x = detection.pos_x
        pos_y = detection.pos_y
        bbox[i:] = detection
    /* template update                                                               */
    new_template = sess.run([templates], pos_x, pos_x,$target_h, target_w, frame$)
    template = (1 -learning_rate) * template + (learning_rate) * new_template
```

5.2 Object Detector training

For our task, we need the object detector to recognize cars from a multitude of angles. Not only on ground level, but also at altitude and overhead. We retrained the YOLO network on two datasets: MIO-TCD for low-angle as in figure 5.2b and COWC for overhead view, as shown in 5.2a.

MIO-TCD This dataset has 786702 images. Divided in two tasks, classification and localization. Eleven classes are labeled. (Articulated truck, Bicycle, Bus, Car, Motorcycle, Motorized Vehicle (i.e. Vehicles that are too small to be labeled into a specific category), Non-motorized vehicle, Pedestrian, Pickup truck, Single unit truck, Work van)

This dataset was used to learn car appearance on low altitude, which is encountered when the drone is transitioning from ground level to higher altitude (car6, car16). Training was done on 1300 epochs.

COWC Contains 32716 images. It consists of high altitude pictures of cars in different contexts (parking lot, highway, street). This dataset was used to learn the overhead appearance of cars that can be encountered when the UAV is at maximum altitude, as in dataset car3 and car9. Training was done on 500 epochs.

5.3 Re-detection

The object detector is triggered when the tracker score is not satisfying. The score has values in $\mathbf{R}$, thus the max value of the score is not a good metric of response quality. Instead, we define Δ, the salience of the score represented by the range of values in the correlation response.

$$\Delta = max - min$$

When Δ drops below a threshold t, we evaluate the object detector on the search area. The best detection is used as a re-initialization input, providing a new score, new target dimensions and a new template.

online update If the max score drops below zero, or if the predicted position is out of the frame, we perform a re-detection over the whole image.

The object detector a new position with a set of coordinates (x, y), and a height w and width w. We use this new prediction as a new reference to generate an augmentation map but also to scale dynamically the search area and bounding box.

```
Algorithm 2: Adaptive Scale Mechanism
  Data: h and w the height and width of the detection, $x\_sz$ the size of the search area,
        target_h and target_w the height and width of our internal representation
  Result: adaptive scale bounding box and search area
  if $target\_h \leq htarget\_w \leq w$:
      $detection = object\_detector(frame)$
      $target_h = h$
      $target_w = w$
  if $x\_sz \leq hx\_sz \leq w$:
      $x\_sz = 1.5 * x\_sz$
```

5.4 Augmentation Map

The Siamese FC relies on the correlation response to determine target position. In our system, we propose to augment the correlation response with the re-detection.

We generate an augmentation score map using the new initialization input provided by the object detector. For each pixel with coordinate (x, y), where x and y belong to the object detector bounding box, we fill the new score map with the maximum value of the existing score map. The augmentation map mimics the desired response we use to train the filter. Basically it is a Dirac signal centered on a new position.

A weighted sum is performed over the new score and the present score, yielding a more accurate response.

5.5 Implementation

Each sequence is stored as images named sequentially in separate folders. Initialization is performed on the first ground-truth, which generates a template. The correlation response threshold parameter t is set at 10 after learning on the datasets. The learning process will be detailed in chapter 6. During re-detection, we perform fast object detection with YOLO within the search area. If loss of track is detected, the re-detection is performed on the whole picture. Output images with rectangular bounding box around the object are stored in a similar fashion as the input images.

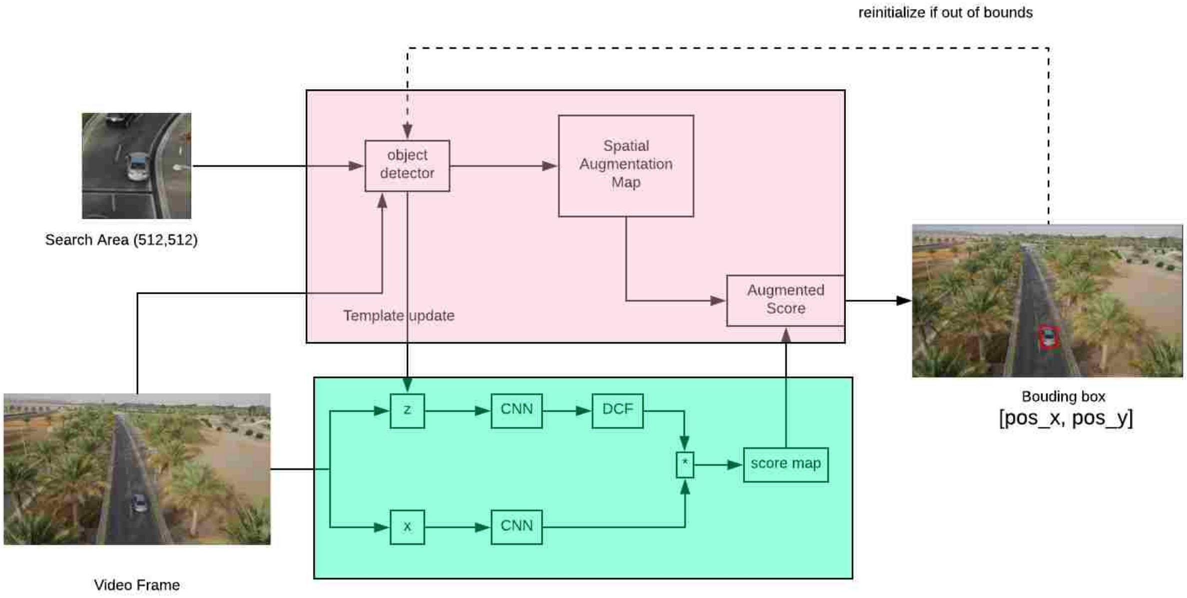
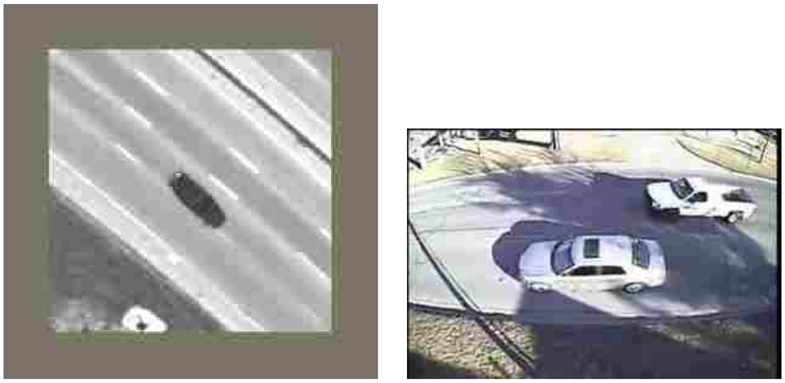


Figure 5.1: Architecture of our proposed model.



(a) COWC

(b) MIO-TCD

Figure 5.2: A comprehensive appearance of the car is learned with 5.2a and 5.2b

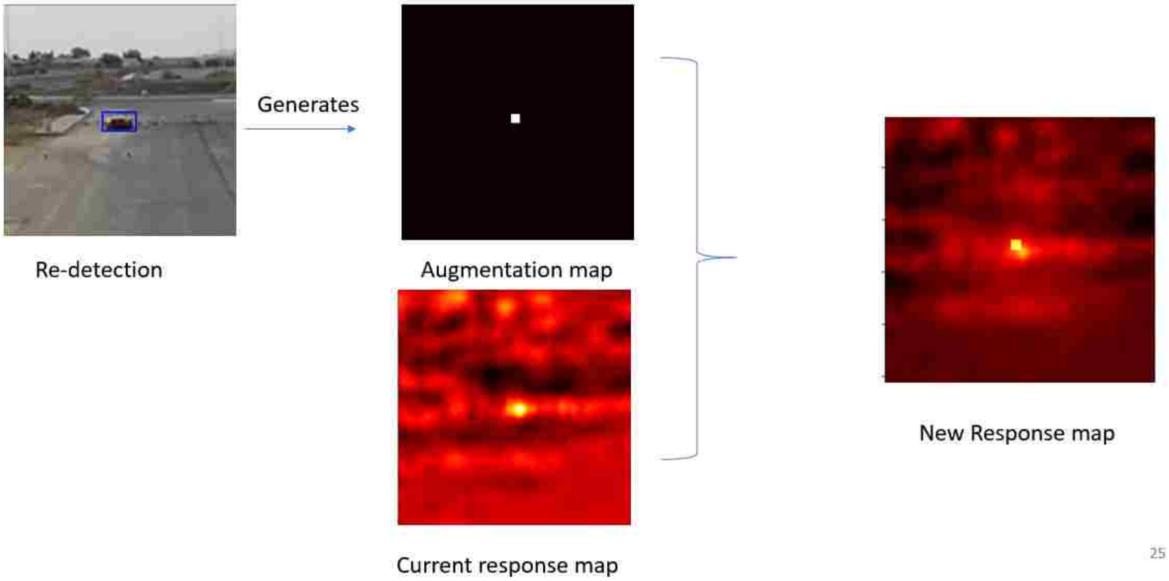


Figure 5.3: Score Augmentation method.

Chapter 6

Experiments and Results

In this section, the preliminary study, the training phase and a comparison between our system versus the original system will be presented. The currently most widely used methodologies developed from three benchmark papers: the Visual Object Tracking challenge (VOT) [?], the Online Tracking Benchmark (OTB) [WLY15] and the Amsterdam Library of Ordinary Videos (ALOV) [SCC+13].

First, we will explain the comparison done with six state-of-the-art trackers performed with the VOT methodology, on public dataset and on classified data. This preliminary study will explain the selection of siameseFC. The trackers evaluated were CSRDCF, CFCF, SiamFC, ECO, CFWCR, gNet.

Then the learning process of our tracker will be explored. First, the detector was retrained, second, we learned a re-detection threshold for our anomaly detection policy.

Finally a quantitative and qualitative comparison is performed, highlighting the performance of our system.

6.1 System Configuration

The proposed algorithm is implemented in Python 2.7 and Tensorflow 1.6 with and in Intel Core i7-3930 CPU @ 3.20Ghz, 64 GB RAM, NVIDIA Titan X, 12 GB VRAM.

The VOT benchmark has been performed with Matlab R2017a, and the object detector has been implemented using C++ and Darknet.

6.2 Datasets

6.2.1 Public Datasets

Our study pertains to long-term ground object tracking using an unmanned aerial vehicle. Available benchmark datasets do not represent our problem, as they are catered to tasks like face tracking, dancer tracking, object tracking, often times with a ground perspective and a fixed point of view. We require a moving elevated camera with moving vehicles to be tracked. Moreover, we focus exclusively on sequences longer than a minute. A comprehensive review was done to find video sequences fulfilling such conditions.

Seven sequences were picked from the UAV123 dataset and Urban Tracker dataset. *car1* (2629 frames), *car3* (1717 frames), *car5* (745 frames) , *car6* (4861 frames) , *car9* (1879 frames) , *car16* (1993 frames), *strene_car22* (802 frames).

Those sequences reflect challenging conditions defined in [WLY13]. Those include low resolution, rapid camera movement, rapid object movement, scale change, rotation, partial/total occlusion and clutter.

6.2.2 Protected Dataset

In addition to public available dataset, a collection of protected dataset was evaluated in the preliminary study. These sequences represent high altitude military grade surveillance sequences and were provided by Toyon.

6.3 Visual Object Tracking Benchmark

The Visual Object Tracking (VOT) toolkit is a MATLAB benchmark tool that evaluates standardized tests on an array of trackers. It is one of the state-of-the-art evaluation benchmark with OTB and .

We performed the baseline benchmark on public and protected dataset. The baseline benchmark is initializing the first frame with the ground truth and let the trackers run until failure. After each failure, the toolkit re-initializes the tracker to the ground truth on the frame where failure happened.

Three performance metrics were evaluated in order to rank the performing trackers. These parameters are:

1. Robustness, with a prediction from the first frame to the last without failure.

2. Accuracy, with minimal error between the ground truth and the tracking result

3. Execution speed, measured in frames per second, to evaluate performance in real-time.

Execution speed is verified easily during run time, by comparing their respective speed. However, comparing the first two criteria is a much more complex task. The VOT evaluation benchmark uses several metrics to perform comparison. They are defined as accuracy, robustness and expected overlap. At time step t, with a predicted bounding box A_t^T and a ground-truth bounding box A_t^G, the overlap Φ_t is:

$$\Phi_t = \frac{A_t^G \cap A_t^T}{A_t^G \cup A_t^T}$$

Accuracy is defined the average overlap between the result bounding box and the ground-truth bounding box.

It is averaged on successful tracking periods, yielding a single average accuracy per frame.

Robustness is evaluated by the number of failures. A failure is defined by the number of times the tracker drifted and had to be reinitialized. A failure is detected when the overlap reached zero.

6.4 Object Tracking Benchmark Benchmark

The OTB approach consists of initializing once with ground-truth from the first and letting the trackers to perform until the end of sequence. Contrary to the VOT approach, there is no re-initialization.

The comparison of tracking performance with the ground-truth is expressed in therms of Precision, Precision Area Under Curve, and Intersection Over Union (IOU).

Precision is defined as the percentage of frames where the euclidean distance between the center of the the ground truth bounding-box and the center of the prediction for the tracked object is below a set threshold. In our test we chose the threshold to be 20 pixels, which is the average dimension of our object.

Precision AUC is defined as the integral of the euclidean distance between the center locations of the tracked target and the ground-truth bounding boxes. It is representation of the tracker overall performance.

IOU is defined as the overlap, and is mathematically formulated as in section 6.3.

6.5 Correlation Threshold Learning

As formulated in chapter 5, we re-initialize our tracker when the response quality drops below a threshold t. When t is high, we will re-initialize more often, but also running the risk of rejecting good tracks. On the contrary, a low threshold will mean we will accumulate more error before restart. Due to the complex and diverse situations encountered with UAV sequences, a balance between a weak and strong re-detection policy has to be determined.

If the threshold is too low, the tracker is most likely to fail without re-initializing. However, if the threshold is too high, we run the risk to rely on a wrong detection for re-initialization. We tested a range of threshold values in order to learn an optimal value. Precision AUC (table 6.1) and intersection over union (table 6.2) were considered as deciding factors. It is apparent that a value of 10 for the correlation response threshold yields the best average result.

Car9, which contains full object occlusion did not perform well when the re-detection policy was too weak. Experimentally at run time, the track was lost after occlusion happened. On the other side of the spectrum, car3, car6 and car16, which contain extreme transformation or clutter did not complete the track due to an aggressive re-detection policy.

6.6 Quantitative Result

Preliminary results: We perform the baseline Visual Object Tracking between the six competing trackers. Test were performed in two runs. Public dataset and protected data were evaluated separately. Results for the public data are compiled in table 6.3.

In summary SiameseFC is better on 3 metrics over 4, it has better accuracy, expected overlap and speed. ECO, CFWCR and CFCF earned first position in robustness.

The evaluation on protected data, that represents a real life use case, confirms the superiority of the siameseFC architecture over the competition. In table 6.5, we can see that SiamFC has better accuracy and expected overlap. In this case, robustness wasn't considered, as all the competing trackers performed equally.

By labeling various sub sequences we can isolate which factors influence tracking performance.

We define camera motion as a rapid change in camera view, object motion as a rapid object movement and occlusion the complete or partial disappearance of an object.

Detailed accuracy in table 6.6 and robustness in table 6.7 show 3 things:

- Occlusion has the most influence in overlap decrease and number of failures
- No tracker performs re-initialization, as occlusion introduces failure in every case.
- Camera motion introduces more drift than the target motion. Meaning that a lot of noise is created by UAV

Summary In conclusion, our preliminary study highlighted the shortcomings of the state-of-the-art trackers in terms of UAV-based tracking.

First, even if object motion and camera motion can cause performance issues, occlusion is the main culprit in the decrease in accuracy and robustness.

Moreover, the test on protected data shows that the public dataset are not a good reflection of real-life situations. The performance decreased in the order of 10, and all trackers have suffered massive failures.

However, Siamese FC was selected as the basis of our system, as it performed with adequate speed, was the best in accuracy and performed above average in 4 of our test sequences out of 6.

updated system vs vanilla We test our system against the vanilla SiamFC architecture. Results are detailed in table 6.4. Our tracker performs significantly better on challenging sequences car6, car9 and car16, while keeping good results in other sequences. In out tests, car1 could not reinitialize properly.

We achieved a dramatic improvement in both Precision and overlap. Car6 is the sequence with the most improvement with a precision jump from 5.33% to 45.66%, an overlap increase from 14.57 to 52.42.

6.7 Qualitative Results

The tracker has to cope with a variation of attributes described in [WLY13], such as illumination variation, scale variation, occusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, object disappearance, clutter and low resolution. Testing has been performed on all the 6 public dataset, and we present and discuss our results in this section.

6.7.1 Rapid Appearance change

Figure 6.1 present our tracker in action with dramatic appearance change, out-of-plane rotation, low resolution and motion blur. Our re-detection scheme improves the internal object representation, resulting in better performance with tracking in subsequent frames.

Without the update policy, the vanilla tracker struggles in keeping a relevant object representation, resulting in significant drift as seen in figure 6.4a and 6.4b.

6.7.2 Occlusion, object disappearance

Occlusion has been determined as the major factor of failures. Our tracker show robustness against occlusion in figure 6.2. Whereas other trackers will fail. Our re-detection algorithm allows the track to be initialized after occlusion, resulting in better performance.

In the other hand the vanilla tracker does not detect a drop in detection quality, and learns the occluder as the new template as shown in figure 6.4c .

In figure 6.3, we show that our tracker tackles successfully complex camera movement and partial object disappearance.

Conclusion After extensive experimentation we found that our tracker exhibited promising result with almost all of the attributes mentioned in [WLY13].

Values of t

	5	6	7	8	9	10	11	12	13	14	15
Car3	34.32	34.32	32.47	34.59	35.35	38.57	37.28	NA	NA	NA	NA
Car6	1.78	1.78	1.78	NA	8.56	12.18	3.6	4.85	3.44	NA	NA
Car9	NA	NA	NA	NA	14.95	14.95	14.95	14.95	14.95	14.95	14.95
Car16	24.22	24.23	12.08	NA	21.40	23.71	22.24	11.5	NA	NA	NA
Strene Car22	33.39	33.39	33.39	33.39	33.39	33.39	33.39	37.84	37.84	38.38	38.38

Table 6.1: Precision AUC on public dataset with different correlation threshold

Values of t

	5	6	7	8	9	10	11	12	13	14	15
Car3	50.45	50.45	46.13	50.77	52.85	59.22	50.66	NA	NA	NA	NA
Car6	13.94	13.94	13.94	NA	20.59	52.42	17.88	17.36	20.78	NA	NA
Car9	NA	NA	NA	NA	29.77	29.77	29.77	29.77	29.77	29.77	29.77
Car16	23.46	24.02	33.27	NA	36.9	48.50	45.71	17.44	NA	NA	NA
StRene22	51.9	51.9	51.9	51.9	51.9	51.9	51.9	46.68	46.68	46.95	46.95

Table 6.2: Untersection over Union on public dataset with different correlation threshold

	Accuracy (rank)	Robustness (rank)	Expected Overlap (%)	Speed (fps)
SiamFC	**2.5**	2.25	**0.4402**	**24**
Gnet	3	2.5	0.3911	10
ECO	2.5	**1.25**	0.3612	9
CSRDCF	2.75	2.5	0.3478	10.1
CFWCR	4.75	**1.25**	0.3386	10.4
CFCF	3.5	**1.25**	0.3347	8.2

Table 6.3: Overall baseline results on public dataset.

	Precision (20 px)		Precision (AUC)		IOU		Speed	
	Vanilla	Ours	Vanilla	Ours	Vanilla	Ours	Vanilla	Ours
Car1	NA	NA	NA	NA	NA	NA	NA	NA
Car3	95.16	71.68	34.32	28.34	50.45	42.60	36.48	12.94
Car6	5.33	45.66	1.78	12.18	14.57	52.42	35.91	13.04
Car9	42.28	64.75	14.95	21.62	29.77	40.39	33.32	12.68
Car16	47.89	79.07	15.92	27.71	16.92	53.88	37.41	13.77
strene_22	100.00	98.50	33.39	36.92	51.90	45.38	34.95	13.83

Table 6.4: OTB benchmark result: Ours vs Vanilla

	Accuracy (rank)	Robustness (rank)	Expected Overlap (%)	Speed (fps)
SiamFC	**3.17**	**1**	**0.0870**	NA
Gnet	8.67	**1**	0.0860	NA
ECO	4.67	**1**	0.0846	NA
CSRDCF	6.67	**1**	0.0713	NA
CFWCR	5.83	**1**	0.0854	NA
CFCF	4.33	**1**	0.0812	NA

Table 6.5: Overall Baseline Results on Toyon Data

	Camera Motion		Motion Change		Occlusion		Weighted Mean	
	Rank	Overlap	Rank	Overlap	Rank	Overlap	Rank	Overlap
SiamFC	3	0.41	4	0.32	**2**	**0.28**	**2.5**	**0.59**
Gnet	1	**0.46**	**1**	**0.51**	4	0.25	3	0.55
ECO	4	0.39	2	0.42	2	0.25	2.5	0.57
CSRDCF	2	0.42	3	0.32	3	0.28	2.75	0.57
CFWCR	6	0.34	6	0.28	6	0.22	4.75	0.55
CFCF	4	0.39	4	0.30	4	0.34	3.50	0.56

Table 6.6: Detailed Accuracy Results

	Camera Motion		Motion Change		Occlusion		Weighted Mean	
	Rank	Failures	Rank	Failures	Rank	Failures	Rank	Failures
SiamFC	**1**	**0**	**1**	**0**	5	2	2.25	1.65
Gnet	2	1	**1**	**0**	6	3	2.5	1.86
ECO	2	1	**1**	**0**	**1**	**1**	1.25	1.76
CSRDCF	6	2	**1**	**0**	**1**	**1**	2.50	2.69
CFWCR	2	1	**1**	**0**	**1**	**1**	1.25	1.76
CFCF	2	1	**1**	**0**	**1**	**1**	**1.25**	**0.98**

Table 6.7: Detailed Robustness Results

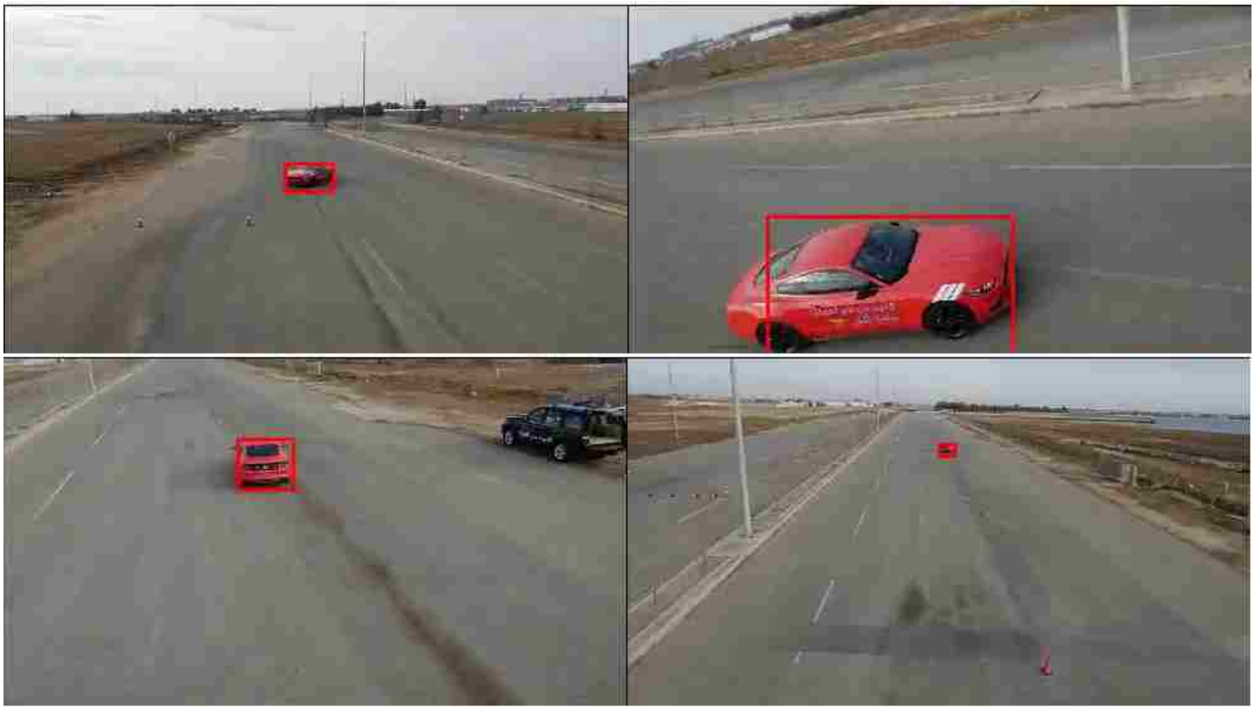


Figure 6.1: Tracking with robust detection with scale variation and high degree of appearance change

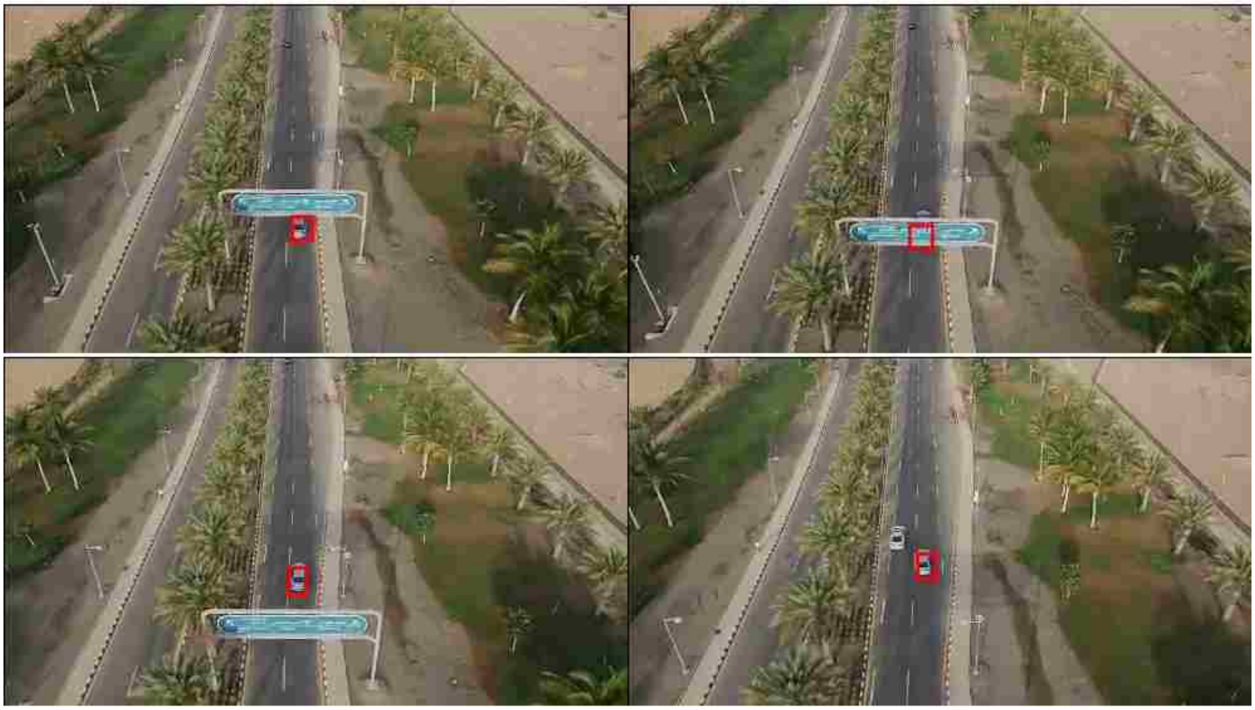


Figure 6.2: Tracking with robust detection with total object disappearance

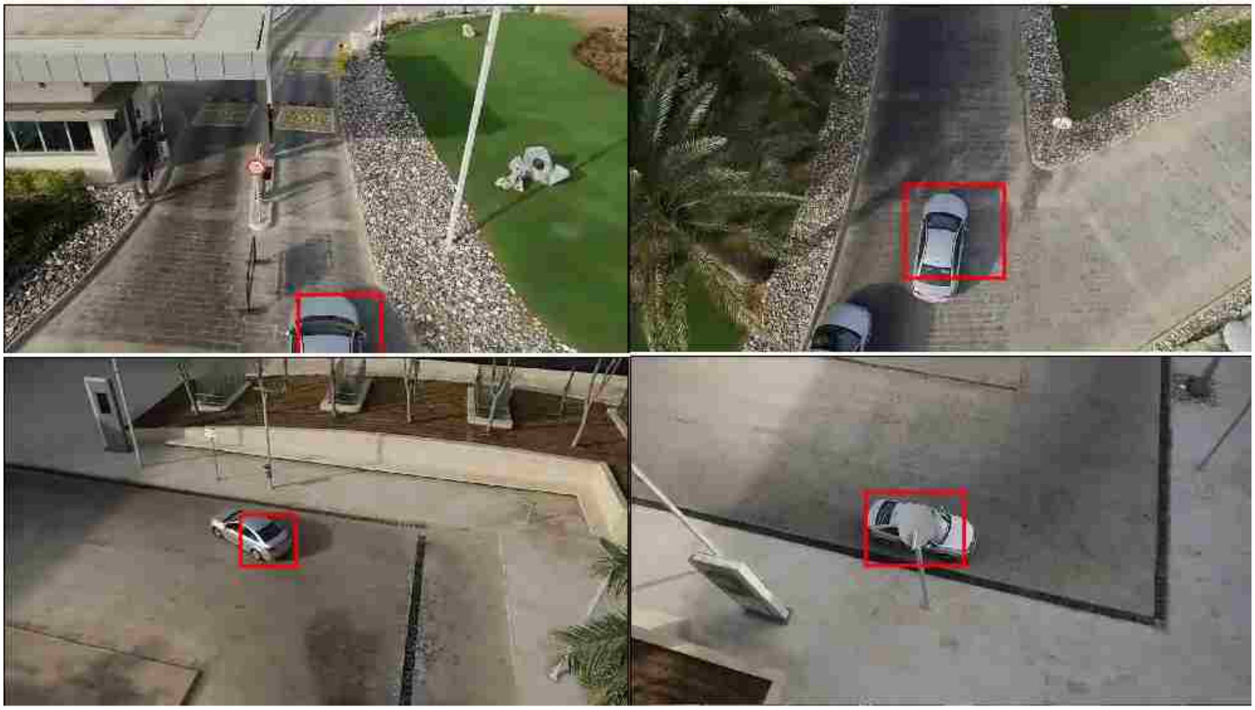
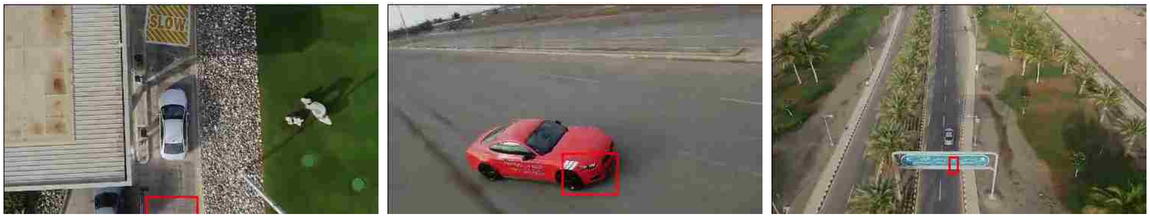


Figure 6.3: Tracking with robust detection with out-of-view and high degree of appearance change



(a) car6: rapid camera motion (b) Car16: rapid object motion (c) car9: occlusion

Figure 6.4: Various failure cases in the vanilla tracker in challenging sequences

Chapter 7

Conclusion and Future Works

7.0.1 Summary

In this thesis, we aimed to develop an aerial tracking system for ground target on long sequences. Comprehensive research has been performed to evaluate highly accurate state-of-the-art object trackers and detectors and the most relevant have been tested. Even if the state-of-the-art trackers performed well on the existing reference benchmarks, they were all poorly suited to our task. Current benchmarks focus on short-term ground to ground tracking, and our evaluation, both on consumer grade data and classified data, has showed the ineptitude to tackle UAS-based tracking.

The Fully connected Siamese networks tracker (siamFC) from [BVH+16] showed good potential, as it was training the classifier in the Fourier Domain, using a regularized distance function between the circular shifts of the image and a Dirac signal, kernelized correlation and circulant matrix resulting in high speed. However, performance was shown to be poor, especially for object disappearance and, in a lesser extent scale variation, due to fast relative motion between camera and object, and partial and total occlusion.

An adaptive re-detection system with real-time anomaly detection was proposed to address failures and shortcomings of the SiamFC, by exploiting the values of the correlation response. Notably, our new system generates an adaptive bounding box around the object with high degree in variation of dimensions and appearance, and re-initializes correctly after object disappearance.

A new score augmentation method was designed to boost the quality of the correlation response generated by the Discriminative Correlation Filter. The propose algorithm was implemented in

Python and Tensorflow.

Following a qualitative and quantitative evaluations on challenging datasets, we found that our system outperforms drastically the other trackers in terms of accuracy, while maintaining a real-time speed. In conclusion, the proposed system successfully demonstrated better tracking than other state-of-the-art trackers.

7.0.2 Future Work

Several limitations came up during our experimentation. We identified several situations where our method fails.

- The tracking robustness relies heavily on the quality of the object detection. Wrong detection results in total failure of the track.
- If the object is completely out-of-view for a long period of time, re-initialization fails. The proposed method can confuse multiple object with similar appearance and assigns a bounding box to the wrong object. (example car1)

It is also important to note that our solution is not fully trained end-to-end, as it relies on a threshold for re-detection. One could imagine a system that evaluates online the quality of the correlation response, resulting in an adaptive threshold. However the question of evaluating the quality of a correlation map is an other field of study. As of now, no exploitation of the numerical values has been developed, and the variation in response is hard to quantify as the appearance of the correlation maps varies a lot.

In order to adapt our tracker for other applications, we would have to retrain our object detector, which is time consuming and resource intensive. This is an issue not specific to our system but to all deep learning and data-driven systems.

Bibliography

[BVH$^+$16] Luca Bertinetto, Jack Valmadre, João F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. *arXiv preprint arXiv:1606.09549*, 2016.

[CLK15] Luka Cehovin, Ales Leonardis, and Matej Kristan. Visual object tracking performance measures revisited. *CoRR*, abs/1502.05803, 2015.

[DBKF16] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ECO: efficient convolution operators for tracking. *CoRR*, abs/1611.09224, 2016.

[GA17] Erhan Gundogdu and A. Aydin Alatan. Good features to correlate for visual tracking. *CoRR*, abs/1704.06326, 2017.

[GDDM14] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[Gir15] Ross Girshick. Fast r-cnn. *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015.

[HFZ$^+$17] Zhiqun He, Yingruo Fan, Junfei Zhuang, Yuan Dong, and Hongliang Bai. Correlation filters with weighted convolution responses. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 1992–2000, 2017.

[IMJ$^+$15] A. Idries, N. Mohamed, I. Jawhar, F. Mohamed, and J. Al-Jaroodi. Challenges of developing uav applications: A project management view. In *2015 International Conference on Industrial Engineering and Operations Management (IEOM)*, pages 1–10, March 2015.

[KLM$^+$17] Matej Kristan, Aleš Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Čehovin Zajc, Tomas Vojir, Gustav Häger, Alan Lukežič, Abdelrahman Eldesokey, and Gustavo Fernandez. The visual object tracking vot2017 challenge results, 2017.

[LAE$^+$16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016.

[LVZ^{+}18] Alan Lukezic, Toms Vojr, Luka Cehovin Zajc, Jiri Matas, and Matej Kristan. Discriminative correlation filter tracker with channel and spatial reliability. *International Journal of Computer Vision*, 126(7):671–688, 2018.

[MSG16] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2016.

[RDGF15] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[RF16] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. *CoRR*, abs/1612.08242, 2016.

[SCC^{+}13] Arnold W. M. Smeulders, Dung Manh Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, and Mubarak Shah. Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36:1442–1468, 2013.

[SSAF^{+}18] Hazim Shakhatreh, Ahmad Sawalmeh, Ala I. Al-Fuqaha, Zuochao Dou, Eyad Almaita, Issa M. Khalil, Noor Shamsiah Othman, Abdallah Khreishah, and Mohsen Guizani. Unmanned aerial vehicles: A survey on civil applications and key research challenges. *CoRR*, abs/1805.00881, 2018.

[ST17] Kartik Umesh Sharma and Nileshsingh V. Thakur. A review and an approach for object detection in images. *Int. J. Comput. Vision Robot.*, 7(1/2):196–237, January 2017.

[VBH^{+}18] Jack Valmadre, Luca Bertinetto, João F. Henriques, Ran Tao, Andrea Vedaldi, Arnold W. M. Smeulders, Philip H. S. Torr, and Efstratios Gavves. Long-term tracking in the wild: A benchmark. *CoRR*, abs/1803.09502, 2018.

[vZLK16] Luka Čehovin Zajc, Aleš Leonardis, and Matej Kristan. Visual object tracking performance measures revisited, Apr 2016.

[WLY13] Y. Wu, J. Lim, and M. Yang. Online object tracking: A benchmark. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2411–2418, June 2013.

[WLY15] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(9):1834–1848, 2015.

[ZZtXW18] Zhong-Qiu Zhao, Peng Zheng, Shou tao Xu, and Xindong Wu. Object detection with deep learning: A review. *CoRR*, abs/1807.05511, 2018.

Curriculum Vitae

Graduate College

University of Nevada, Las Vegas

Marc Dinh

Degrees:

Bachelor of Science in Mathematics and Computer Science 2013

University of Marne-la-vallee

Thesis Title: UAS-BASED OBJECT TRACKING VIA DEEP LEARNING

Thesis Examination Committee:

Chairperson, Yoohwan Kim, Ph.D.

Committee Member, Justin Zhan, Ph.D.

Committee Member, Fatma Nasoz, Ph.D.

Graduate Faculty Representative, Brendan Morris, Ph.D.