

*Deep Learning Project Report*

**Person Tracking System**

By

Sandhi Jain (209309030)

**Under the supervision**

**of:**

Rohini R. Rao

Abhilash K. Pai

**Department of Data Science and Computer Applications**

**Faculty of Engineering**

**Manipal Institute of Technology**

**Manipal**

October 2022

Table of Contents:

1. **Abstract**  3
2. Introduction ……………………………………………………….4
3. Theoretical Background………………………………………….5
4. Project Objectives……………………………………………….19
5. Methodology……………………………………………………..19
6. Results……………………………………………………………
7. Discussion…………………………………………………………..
8. Conclusion and Future Research…………………………………
9. Literature…………………………………………………………….

Abstract:

Multiple people tracking is a key problem for many applications such as surveillance, animation or car navigation, and a key input for tasks such as activity recognition. In crowded environments occlusions and false detections are common, and although there have been substantial advances in recent years, tracking is still a challenging task. Tracking is typically divided into two steps: detection, i.e., locating the pedestrians in the image, and data association, i.e., linking detections across frames to form complete trajectories. For the data association task, approaches typically aim at developing new, more complex formulations, which in turn put the focus on the optimization techniques required to solve them. However, they still utilize very basic information such as distance between detections. In this thesis, I focus on the data association task and argue that there is contextual information that has not been fully exploited yet in the tracking community, mainly social context and spatial context coming from different views.

This involves the following steps:

* Detection:- First, all the objects are detected in the frame.
* Association:- Once we have detection for the frame, a matching is performed for similar detections with respect to the previous frame.
* Object detection:- detects if there are any objects in an image. (eg. x and y coordinates & width and height)
* Object recognition:- recognizes what kind of the detected objects are (eg. class labels)
* Object Tracking:- it is the process of locating moving objects over time in videos. It involves the following steps:

1. Taking an initial set of object detections,
2. Creating unique ID for each of the detections,
3. Tracking the objects over time,
4. Maintaining the ID assignment.

Dataset:

We aim to use a combination of the datasets:

Dataset used for this project:

[STEP-ICCV21](https://motchallenge.net/data/STEP-ICCV21/) - The [Segmenting and Tracking Every Pixel (STEP)](https://arxiv.org/abs/2102.11859) benchmark consists of 2 training sequences and 2 test sequences. It is based on the MOTChallenge and Multi-Object Tracking and Segmentation (MOTS) benchmark. This benchmark extends the annotations to the STEP task. To this end, we added dense pixel wise segmentation labels for every pixel. In this benchmark, every pixel has a semantic label and all pixels belonging to the most salient object class, pedestrian, have a unique tracking ID. We evaluate submitted results using the [Segmentation and Tracking Quality (STQ)](https://arxiv.org/abs/2102.11859) metric. This benchmark is part of the ICCV21-Workshop [Segmenting and Tracking Every Point and Pixel](https://motchallenge.net/workshops/bmtt2021/).

[MOT20](https://motchallenge.net/data/MOT20/) - Pedestrian Detection Challenge. This benchmark contains 8 challenging video sequences (4 train, 4 test) in unconstrained environments. Tracking and evaluation are done in image coordinates. All sequences have been annotated with high accuracy, strictly following a well-defined protocol.

## **Introduction:**

## What is Object Tracking?

Object tracking is a deep learning process where the algorithm tracks the movement of an object. In other words, it is the task of estimating or predicting the positions and other relevant information of moving objects in a video.

Object tracking usually involves the process of object detection. Here’s a quick overview of the steps:

* Object detection, where the algorithm classifies and detects the object by creating a bounding box around it.
* Assigning unique identification for each object (ID).
* Tracking the detected object as it moves through frames while storing the relevant information.

## 

## Object Tracking vs. Object detection

Object tracking refers to the ability to estimate or predict the position of a target object in each consecutive frame in a video once the initial position of the target object is defined.

On the other hand, object detection is the process of detecting a target object in an image or a single frame of the video. Object detection will only work if the target image is visible on the given input. If the target object is hidden by any interference it will not be able to detect it. Object tracking is trained to track the trajectory of the object despite the occlusions.

### **Theoretical Background**

### **Types of Object Tracking**

There are two types of object tracking: image tracking and video tracking.

#### **Image tracking**

Image tracking is the task of automatically recognizing and tracking the images.

It is mostly applied in the field of augmented reality (AR). For instance, when given a two-dimensional image as an input through a camera, the algorithm detects two-dimensional planar images, which can be then used to superimpose a 3D graphical object.

Once the 3D graphic is superimposed, the user can move the camera without actually losing track of the 2D planar surface and graphic on top of it.

Companies like Apple and Ikea use such technologies to give the customers a virtual experience of how their products will look in their personal settings.

#### **Video tracking**

Video tracking is the task of tracking a moving object in a video.

The idea of video tracking is to associate or establish a relationship between target objects as it appears in each video frame. In other words, video tracking is analyzing the video frames sequentially and stitching the past location of the object with the present location by predicting and creating a bounding box around it. Video tracking is widely used in traffic monitoring, self-driving cars, and security because it can process real-time footage.

4 stages of the Object Tracking process

**Target initialization**

The first step involves defining the object of interest or targets.

It incorporates the process of drawing a bounding box around it in the initial frame of the video. The tracker must then estimate or predict the object’s position in the remaining frames while simultaneously drawing the bounding box simultaneously.

### **Appearance modeling**

Appearance modeling deals with modeling the visual appearance of the object. When the targeted object passes through various scenes like the lighting condition, angle, speed, etc., they may change the appearance of

the object, and it may lead to misinformation and the algorithm losing track of the object.

Appearance modeling has to be conducted so that modeling algorithms can capture various changes and distortions introduced when the target object moves.

Appearance modeling consists of two components:

1. Visual representation: it focuses on constructing robust features and representation that can describe the object
2. Statistical modeling: it uses statistical learning techniques to build mathematical models for object identification effectively.

### **Motion estimation**

Motion estimation usually infers the predictive capability of the model to predict the object’s future position accurately.

### **Target positioning**

Motion estimation approximates the possible region where the object could most likely be present. Once the location of the object is approximated, we can then use a visual model to lock down the exact location of the target.

## Levels of Object Tracking

Object tracking can be defined by two levels:

1. Single Object Tracking(SOT)
2. Multiple Object Tracking(MOT): it aims to track objects of multiple classes as we see in self-driving cars.

### 

### **Single Object Tracking**

Single Object Tracking (SOT) aims to track an object of a single class instead of multiple objects. It is also sometimes referred to as *Visual Object Tracking.*

In SOT, the bounding box of the target object is defined in the first frame. The goal of the algorithm is then to locate the same object in the rest of the frames.

SOT belongs to the category of detection-free tracking because one has to manually provide the first bounding box to the tracker. This means that

Single Object Trackers should be able to track whatever object they are given, even an object on which no available classification model was trained.

### **Multiple Object Tracking**

Multiple Object Tracking (MOT) refers to the approach where the tracking algorithm tracks every single object of interest in the video.

Initially, the tracking algorithm determines the number of objects in each frame, following that it keeps track of each object’s identity from one frame to the next frame until they leave the frame.

## Object tracking challenges (and solutions)

There are several challenges that one might face while working on object tracking algorithms.

Firstly, it is easy to track an object on a straight road or in a simple environment. In a real-world scenario, the target object will go through deformation, occlusion, background noise, etc.

Occlusion

The occlusion of objects in videos is one of the most common challenges. It refers to an interference phenomenon where the object is affected by the background or foreground in which the tracking algorithm loses track of the object. In other words, the algorithm gets confused as multiple objects come

closer. This leads to the issue where the initially identified object is again (mistakenly) tracked as a new object.

One can implement occlusion sensitivity to prevent it. Occlusion sensitivity allows the user to identify which particular feature of the object is confusing the network. Once identified, similar images can be used to correct the biases and help the network to extract features that differentiate the objects.

### Background clutter

In any machine learning or deep learning task, the background of the images fed into the algorithm creates a lot of issues. It is the same with object tracking models.

In theory, the more densely populated the background, the more difficult it is to extract features, detect or even track the object of interest.

A densely populated background introduces redundant information or noise that makes the network less receptive to features that are important; they also make the network slow to learn and optimize.

To prevent background clutter, one can use a well-curated dataset that has a sparse background.

### Training and Tracking Speed

The modern deep learning algorithms have become much more complex, which means they can extract features and make meaningful correlations; this, in turn, also means that they consume more energy and time.

The tracking algorithm is not a single task algorithm, as we see in image classification and object detection. It is a multitask algorithm that performs object detection, localization, classification and also keeps track of the objects. This type of algorithm is mathematically complex, and it takes a lot of time to train.

We must also keep in mind that the tracking algorithms must perform quickly during the inference time to yield accurate results. Enhancing tracking speed is especially imperative for real-time object tracking models.

To enable faster inference time, the model needs to be carefully designed or chosen. A convolutional neural network, which is the primary architecture for computer vision tasks, can be used in object tracking. These networks are capable enough to perform with great accuracy if designed carefully.

Some of the common algorithms used for tracking objects are Fast R-CNN and Faster R-CNN and their variants. These networks have been proven very efficient in the task of object tracking.

### Varying spatial scales

### One of the issues with object tracking is that the target objects can vary in shape and size; with such a variety of information, the learning algorithm can get confused, leading to generalization error.

Below is the list of techniques that can help to tackle the issue of varying spatial scales:

#### Anchor Boxes

These are the predefined measurement of the target object. The boxes are meant to acquire the scale and aspect ratios of target objects. These boxes are fed into the network during training, and it allows the network to learn and understand the position and size of the object.

Anchor boxes also allow the network to detect multiple objects if they are overlapping by separately evaluating features and yielding accurate results.

#### Feature extraction

Features extraction is an important process in all deep learning techniques.

It enables neural networks to understand the data fed into them. The convolutional neural network (CNN) used for object tracking and other computer vision tasks can efficiently extract spatial information.

These networks, however, must be able to extract multi-scale spatial information. At times, when the object is smaller in the given input image, the network may lose too much signal during downsampling or as the signal propagates through the network. As a result, the network will not be able to detect and track smaller objects during the inference. To tackle such a situation, one should use or develop a network that can preserve the information extracted from each layer so that the network can detect objects within multiple CNN layers, including earlier layers where higher resolution remains.

## Deep Learning-based approaches to Object Tracking

## Object tracking has been around for almost 20 years now and a lot of methods and ideas were introduced to improve the accuracy and efficiency of the tracking models.

Some of the methods involved traditional or classical machine learning approaches like k-Nearest Neighbor or Support Vector Machine. These approaches are good in predicting the target object, but they require important and discriminatory features extracted by professionals.

On the other hand, deep learning methods extract these important features and representations by themselves.

MDNet

Multi-Domain Net is a type of object tracking algorithm which leverages large-scale data for training. Its objective is to learn vast variations and spatial relationships. MDNet is trained to learn the shared representation of

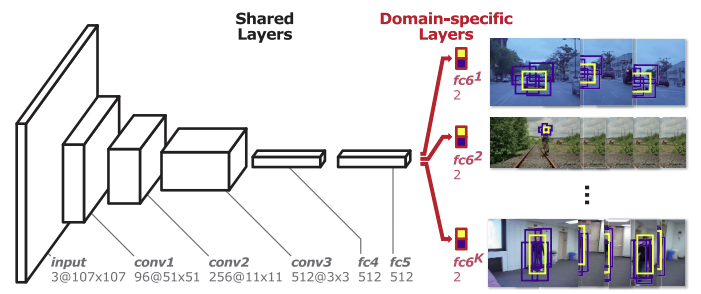
targets from multiple annotated videos, meaning it takes multiple annotated videos belonging to different domains.

MDNet consists of pretraining and online visual tracking:

Pretraining: In pretraining, the network is required to learn multi-domain representation. To achieve this, the algorithm is trained on multiple annotated videos to learn representation and spatial features.

Online visual tracking: Once pre-training is done, the domain-specific layers are removed and the network is only left with shared layers, which consist of learned representations. During the inference, a binary classification layer is added, which is trained or fine-tuned online.

This technique saves time as well as it has proven to be an effective online-based tracking algorithm.

[*Source*](https://arxiv.org/pdf/1510.07945.pdf)

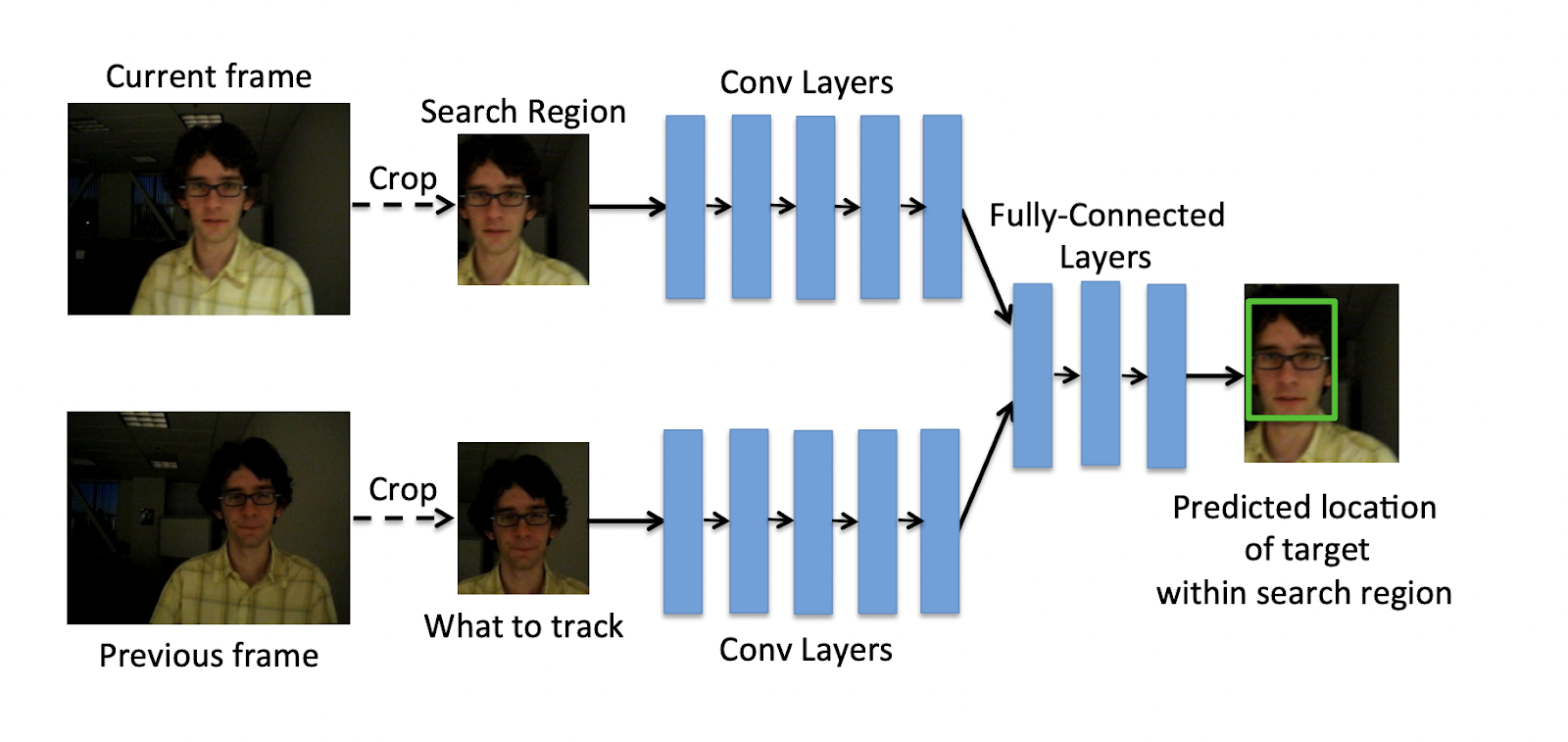
### GOTURN

Deep Regression Networks are offline training-based models. This algorithm learns a generic relationship between object motion and appearance and can be used to track objects that do not appear in the training set.

Online tracker algorithms are slow and do not perform well in real-time; this is because they cannot take advantage of a large number of videos to improve their performance. Offline tracker algorithms, on the other hand, can be trained to handle rotations, changes in viewpoint, lighting changes, and other complex challenges.

Generic Object Tracking Using Regression Networks or GOTURN uses a regression-based approach to tracking objects. Essentially, they regress directly to locate target objects with just a single feed-forward pass through the network.

The network takes two inputs: a search region from the current frame and a target from the previous frame. The network then compares these images to find the target object in the current image.

[*Source*](https://davheld.github.io/GOTURN/GOTURN.pdf)

### ROLO—Recurrent YOLO

ROLO is a combination of recurrent neural networks and YOLO. Generally, LSTM is preferred in combination with CNN.

ROLO combines two types of neural networks: one is CNN which is used to extract spatial information while the other is an LSTM network which is used for finding the trajectory of the target object.

At each time step, spatial information is extracted and sent to the LSTM, which then returns the location of the tracked object.

[*Source*](https://arxiv.org/pdf/1607.05781v1.pdf)

We can use the above diagram to understand how ROLO works:

1. The video sequence is fed into the YOLO architecture which is primarily made of CNN, here features are extracted as well as bounding boxes are detected.
2. The visual features and bounding boxes are then concatenated and fed to the LSTM

3.The LSTM then predicts the trajectory of the objects.

### DeepSORT

DeepSORT is one of the most popular object tracking algorithms. It is an extension to Simple Online Real-time Tracker or SORT, which is an online-based tracking algorithm.

SORT is an algorithm that uses the Kalman filter for estimating the location of the object given the previous location of the same. The Kalman filter is very effective against the occlusions.

SORT comprises of three components:

1. Detection: Detecting the object of interest in the initial stage i.
2. Estimation: Predicting the future location i+1 of the object from the initial stage using the Kalman filter. It is worth noting that the Kalman filter just approximates the object’s new location, which needs to be optimized.
3. Association: As the Kalman filter estimates the future location of the object i+1, it needs to be optimized using the correct position. This is usually done by detecting the position of the object in that position i+1. The problem is solved optimally using the Hungarian algorithm.

With the basics of SORT out of the way, we can incorporate deep learning techniques to enhance the SORT algorithm. Deep neural networks allow

SORT to estimate the object’s location with much higher accuracy because these networks can now describe the features of the target image.

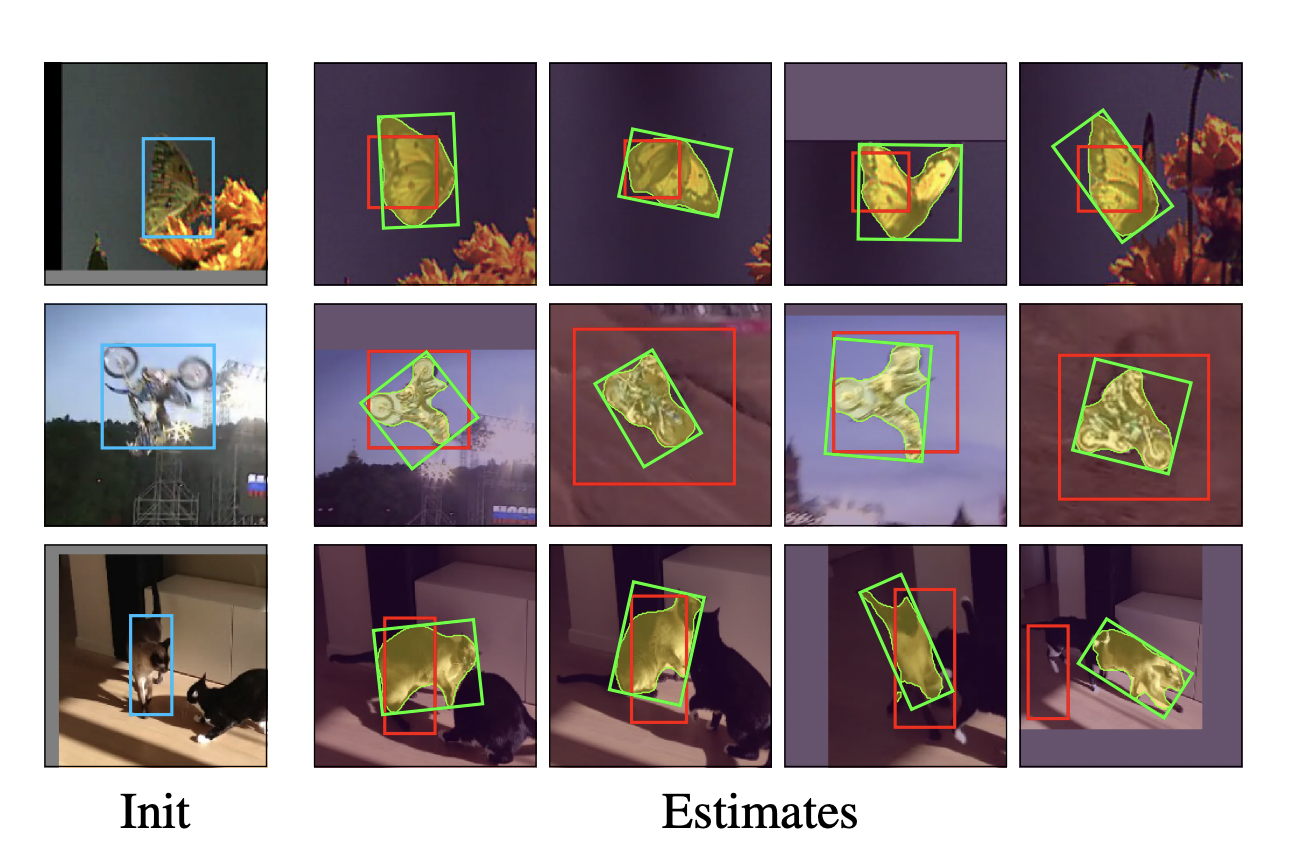
Essentially, the CNN classifier is trained on a task-specific dataset until it achieves good accuracy. Once it is achieved, the classifier is stripped, and

we are left with only the features extracted from that dataset. This extracted feature is then incorporated with the SORT algorithm to track objects.

### SiamMask

SiamMask aims to improve the offline training procedure of the fully-convolutional Siamese network. Siamese networks take in two inputs: a cropped image and a larger search image to obtain a dense spatial feature representation.

The Siamese network yields one output. It measures the similarity of two input images and determines whether or not the same objects exist in the two images. This framework is very efficient for object tracking by augmenting their loss with a binary segmentation task.

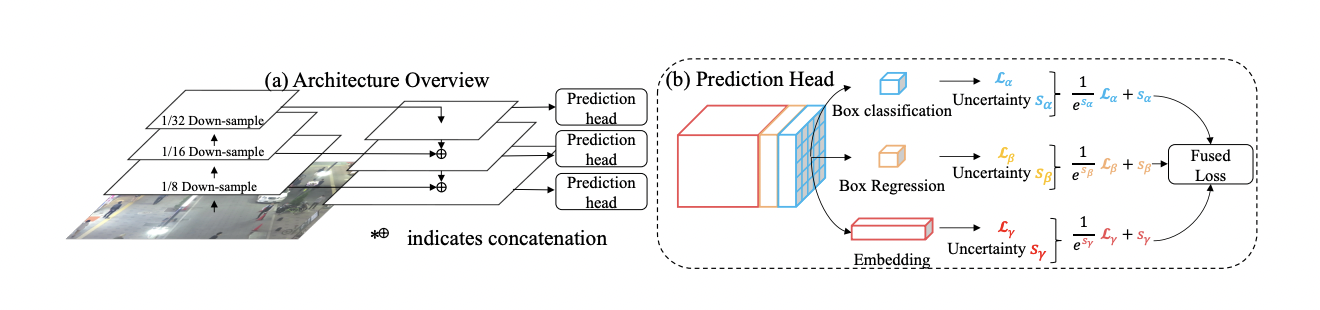
*SiamMask*

Once trained, SiamMask solely relies on a single bounding box initialization and operates online yielding object segmentation masks.

### JDE (Joint Detection and Embedding)

​​Joint Detection and Embedding (JDE) is a single-shot detector designed to solve a multi-task learning problem. JDE learns target detection and appearance embedding in a shared model.

JDE uses Darknet-53 as the backbone to obtain feature representation at each layer. These feature representations are then fused using up-sampling and residual connections. The prediction heads are then attached on top of the fused feature representation, which yields a dense prediction map.

*JDE architecture*

To perform object tracking, JDE yields bounding boxes classes and appearance embedding from the prediction head. These appearance embeddings are compared to embeddings of previously detected objects using an affinity matrix.

Finally, the Hungarian algorithm and the Kalman filter are used for smoothing out the trajectories of the target object and as well as estimating the locations of the same.

### Tracktor++

Tracktor++ is an online tracking algorithm. It uses an object detection method to perform tracking by training a neural network only on the task of detection.

It essentially predicts the position of an object in the next frame by calculating the bounding box regression. It doesn't perform any training or optimization on tracking data.

The object detector for Tracktor++ is usually Faster R-CNN with 101-layer ResNet and FPN. It extracts features from the current frame by using the regression branch of Faster R-CNN.

Project Objectives:

* Person detection is necessary and critical work in any intelligent video surveillance framework, as it gives the essential data to semantic comprehension of the video recordings.
* Compare performance of various object tracking algorithms (DeepSORT, SORT, MD-Net, OpenCV trackers) and object detection algorithms ( Faster R-CNN, YOLOv1, YOLOv2, YOLOv3, Tiny YOLO and Single Shot Detector)
* Aim is to train the model to deal with occlusions, background clutter and improve training and tracking speed.

Methodology:

Optimizing the videos to make it as light as possible for the

algorithm to work smoothly and achieve speed. A video is nothing

but a sequence of images, hence achieving the objective of identifying

the object at the lowest-most level of an image will result in achieving

the same on a video by iterating the process on all the images that the

video is made up of.

Optimization includes :

* Turning the image/video into grayscale
* Reducing the image/video to the pure black white form
* Masking the image/video