Visual Object Tracking

Param Dave
SCIA - 2024

Contents

1 Introduction

This project is an implementation of a visual object multi-tracking using numerous techniques.

This project is divided into multiple components, each are algorithmic improvements added.

In this report, we will dwelve into these different components and their use.

We tested our multi object tracker on a unique video with the detections frame by frame provided in form of bounding boxes and a confidence score.

Our goal is to create and asssociate a track to each of these detections.

The code is in the src/ folder. You can also find their the code's output.
You can find the workflow of the different blocks of implementation in the annexe.

2 Multi-object tracker

In this section we will explain how we handle multiple tracks at the same time.

For each frame, we get its detections. We then compute a cost matrix for all combination of existing tracks and detections in the specific frame. The details of the computation of the cost matrix will be discussed later in the report.

Once we have the cost matrix, we apply the hungarian algorithm to associate to each track one detection.

Each track is associated to a kalman filter, this kalman filter will discuss the kalman filter later in the report.

For these pairs of tracks and detections that were associated together, we update the kalman filter and we append the detection to the track. We also associate the track id to the detection.

Once this is done for all pairs, we delete all unmatched tracks. Additionally, for all detections that were not paired up with any tracks, we create a new track, initialize its kalman filter and append the detection to this newly created track.

3 Kalman Filter

A Kalman filter is an algorithm used for recursive estimation and control. It is particularly useful for tracking the state of a system over time, even when the measurements are noisy or incomplete.

In the context of multi-object tracking, a Kalman filter can be used to predict the next state of an object based on its current state and a set of measurements. The filter maintains an estimate of the object's state, which includes information such as position, velocity, and size. As new measurements become available, the Kalman filter updates its estimate, providing a smooth and accurate tracking result.

There are as many kalman filters as there are tracks. Indeed, each track is a system thats movement will be estimated via the kalman filter.

3.1 Initialize

The Kalman filter is initialized when a new track is created. It is initialized with the bounding box of the detection.

3.2 Predict

The predict method simply predicts the system's future coordinates. This prediction is used during the computation of the cost matrix, as this prodiction will be compared with each detection in order to get the cost. More the prediction is similar to the bounding box, less the cost will be great.

3.3 Update

The update method is used once we have associated a new detection to a track. This incorporates measurements to correct the predicted state, improving the estimate.

4 Cost matrix

The cost matrix is used to evaluate a specific cost for each combination of tracks and detections. The Hungarian algorithm is then used to pair up a detection and a track.

Multiple caracteristics are used to compute this cost.

4.1 Intersection over Union (IoU)

It is a metric commonly used in computer vision and image processing to evaluate the accuracy of an object detection algorithm. IOU is also known as the Jaccard Index.

The IOU is calculated by finding the intersection area between the predicted bounding box and the ground truth bounding box, and then dividing it by the union area of the two boxes.

In the context of object detection, a higher IOU indicates better overlap between the predicted and previous bounding boxes, suggesting a more accurate detection. It is often used as one of the evaluation metrics for tasks such as object detection, instance segmentation, and multi-object tracking.

4.2 Visual information

We also compute the visual difference between both detected bounding box and previous bounding box.

This is useful as instead of relying solely on geometrical information, the cost will also be affected by the visual differences between both bounding boxes.
We use a machine learning approach by using a model to extract the features of bounding boxes and compare their differences.

We resize the bounding boxes to the same dimensions before feeding them to the model.

The model selected for this task is an efficientNet to minimize the computation time.

This difference, once normalized, is an important component of the cost.

4.3 Bounding Box Sizes

To enhance the robustness of our tracking approach, we introduce a size difference component to the cost matrix.

If the size difference between the last bounding box and a candidate bounding box exceeds a predefined threshold, a penalty is applied to the cost, discouraging associations with significantly varying sizes.

5 Improvements and challenge

With juste the geometrical information, our muti object tracker performs poorly. On average, a track's duration is approximatively **3 frames**. As many bounding boxes overlap, it is difficult to keep a stable track for an object.

This problem disappears as we introduced visual information for the cost. We have an average of **10 frames**.

It is important to note that the track average frame is greatly hindered by the tracks that appear only for one frame or so which can be explained by the detections' low confidence.

Other models have been used, such as **ResNet50** or **MobileNetv2**. The current model is as fast at MobileNetv2.

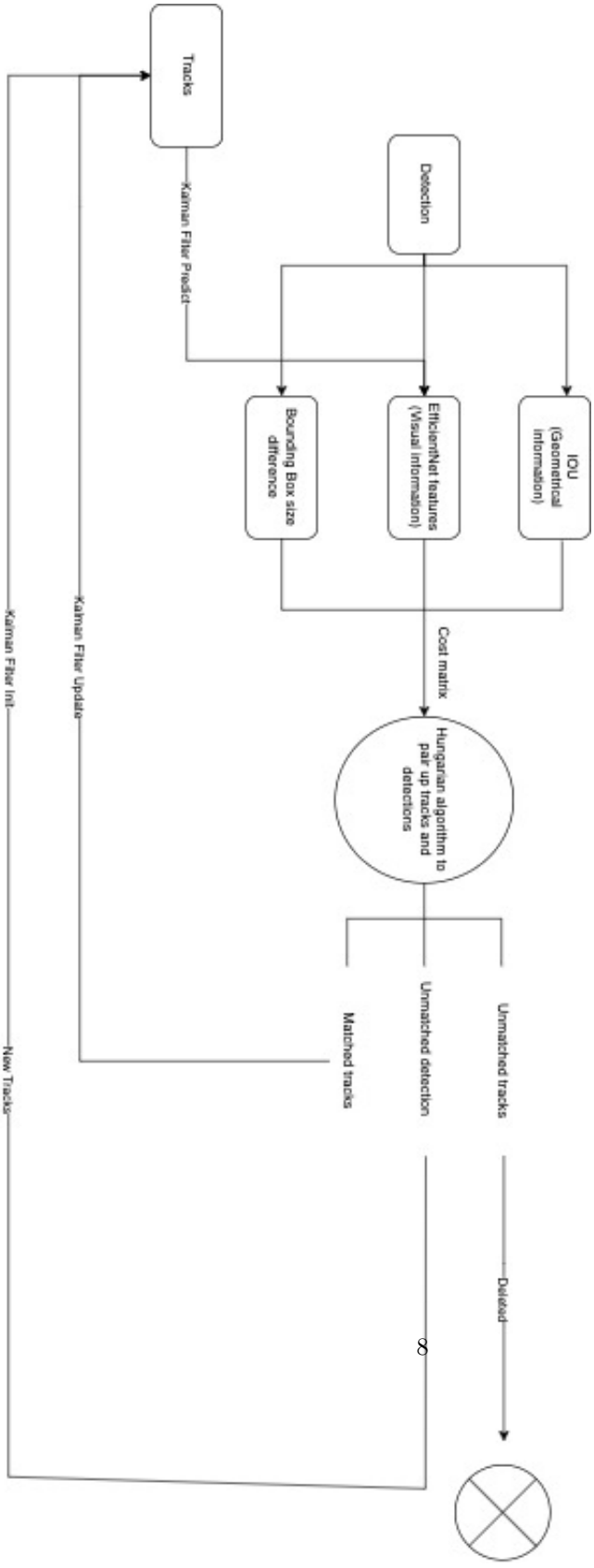
The biggest problem is the time consumption of the model's prediction. Indeed, the calls to the model to extract features of each bounding boxes.

To optimize the time efficiency for each call to the model, we've implemented parallelization in the process of computing the cost matrix. This helps greatly however a lot of time is still needed to process the video.

Another solution would be to keep the calls in cache to minimize the number of calls as much as possible.

I encountered some challenges that hindered the integration of the TrackEval script. Unfortunately, due to compatibility issues between my system configurations and TrackEval dependencies, I faced difficulties in configuring the necessary directories and data paths.

6 Annexe


Tracks
Kalman Filter Predict
Detection
IOU (Geometrical information)
EfficientNet features (Visual information)
Bounding Box size difference
Cost matrix
Hungarian algorithm to pair up tracks and detections
Unmatched tracks
Unmatched detection
Matched tracks
Deleted
Kalman Filter Update
Kalman Filter Init
New Tracks