sciendo

VOLUME 17, N° 1 2023
Journal of Automation, Mobile Robotics and Intelligent Systems

People Tracking in Video Surveillance Systems Based on Artificial Intelligence

Submitted: 15th October 2022; accepted: 2nd January 2023

Abir Nasry, Abderrahmane Ezzahout, Fouzia Omary

DOI: 10.14313/JAMRIS/1-2023/8

Abstract:
As security is one of the basic human needs, we need security systems that can prevent crimes from happening. In general, surveillance videos are used to observe the environment and human behavior in a given location. However, surveillance videos can only be used to record images or videos, without additional information. Therefore, more advanced cameras are needed to obtain other additional information such as the position and movement of people. This research extracted this information from surveillance video footage using a person tracking, detection, and identification algorithm. The framework for these is based on deep learning algorithms, a popular branch of artificial intelligence. In the field of video surveillance, person tracking is considered a challenging task. Many computer vision, machine learning, and deep learning techniques have been developed in recent years. The majority of these techniques are based on frontal view images or video sequences. In this work, we will compare some previous work related to the same topic.

Keywords: *Person tracking, Person detection, Person identification, Video surveillance, Artificial intelligence.*

1. Introduction

Nowadays, video surveillance is expanding rapidly, both technologically and economically. It has become one of the essential links in the security policies of governments. This evolution responds to the security needs of every citizen, in line with the increase of delinquency and criminality.

Video surveillance is now becoming increasingly necessary to monitor both public and private places. In this context, camera networks are installed in abundance in the streets, shopping centers, public transportation, offices, airports, apartment buildings, etc.

A video surveillance system consists essentially of monitoring a multiple number of security camera feeds at the same time. However, the increase in the number of installed cameras makes it extremely difficult to manually process the data generated by these cameras. To help security monitoring personnel explore this data, it is necessary to make the video surveillance task by automating some of its functions. Among these include object detection, person detection, event and human action recognition, tracking of people, etc. Another application is to recognize people who leave the field of view of one camera and reappear in another. The video surveillance system must then be able to reidentify the person and continue the tracking.

Figure 1. Casablanca is giving itself the means to fight against insecurity

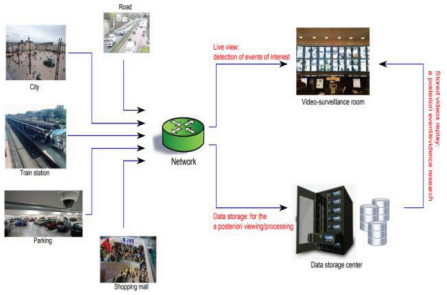

Figure 2. Video-surveillance architecture: live viewing and a posteriori viewing

The research of the estimation of the 3D movement of a person is an important field of computer vision, because of its numerous possible applications: human–computer interfaces, animation, interaction with virtual environments, games, etc.

Capturing 3D human motion in real time, with a single or multiple cameras and without markers is a difficult to achieve. This is due to the ambiguities resulting from the lack of information of depth, partial occultation of human body parts, the elevated number of degrees of freedom, and the variation in the proportions of the human body, as well as the color of the clothes of the different people present in the scene. For these reasons, the number of works dealing with the estimation of people tracking continues to increase.

In this paper, we will present the different approaches to pose estimation and the tracking of people's movements.

Figure 3. Object tracking in deep learning

2. Methods

2.1. Definition

Object tracking is an application of deep learning in which the program takes an initial set of object detections and develops a unique identification for each of the initial detections, and then tracks the detected objects as they move through frames of a video.

In other words, object tracking involves automatically identifying objects in a video and interpreting them as a set of trajectories with high accuracy. Often there is an indication around the tracked object, for example, a square that follows the object, showing the user where the object is on the screen.

Different type of object tracking

Object tracking is used in a variety of use cases involving different types of input images. Whether the intended input is an image or video, or real-time video versus prerecorded video, it impacts the algorithms used to create object-tracking applications.

The kind of input also impacts the category, use cases, and applications of object tracking. Here, we will briefly describe a few popular uses and types of object tracking, such as video tracking, visual tracking, and image tracking.

Video tracking: Video tracking is the process of locating a moving object (or multiple objects) over time using a camera. It has many uses, including: human–computer interaction, security and surveillance, video communication and compression, augmented reality, traffic control, medical imaging, and video editing. Video tracking can be a time-consuming process due to the amount of data contained in the video. Adding to the complexity is the potential need to use object recognition techniques for tracking, a difficult problem in itself.

Image tracking: Image tracking is intended to detect two-dimensional images of interest in a given input. These images are then continuously tracked as they move through the scene. Image tracking is ideal for datasets with high contrast images (e.g., black and white), asymmetry, few patterns, and multiple identifiable differences between the image of interest and other images in the set. Image tracking relies on computer vision to detect and augment images after the image targets have been predetermined.

Visual tracking: Visual tracking or visual target tracking is a research topic in computer vision that is applied in a wide range of everyday scenarios. The goal of visual tracking is to estimate the future position of a visual target that has been initialized without the availability of the rest of the video.

3. Literature Review

This section provides a short outline of various algorithms employed in the literature. The section is categorized into ancient generic, machine learning, features, and deep learning-based ways. A comprehensive survey of various tracking methods are often found in previous studies.

The main purpose of tracking techniques is to detect objects in a video object in a video sequence and to keep track of the successive images in order to find the trajectories of each detected object. Conventional techniques are generally based on motion and observation models. The motion model involves detecting and predicting the object's location the appearance of the object and its position in the image. Some researchers have used the model-based method for object tracking. Many researchers have used machine learning for object tracking, which classifies the tracked object, such as boosting, random forest, Hough forest, structural learning, and support vector machine. Some have proposed feature-based tracking methods, such as Haar-type features, local binary model, histogram of oriented gradient, scale-invariant feature transform, discrete cosine transform, and shape features [4–7]. Other techniques use Kalman filters or the Hungarian algorithm. In order to improve the performance of the tracking methods, different researchers have combined the information from several indices and presented object tracking methods that combine a feature-based detector with the probabilistic segmentation method. The majority of these methods are mainly developed for frontal view datasets that may suffer from occlusion problems.

3.1. Reviewing Some Related Work

Review on: Convolutional Neural Network–Based Person Tracking Using Overhead Views

This paper emphasizes on overhead view person tracking using Faster region convolutional neural network (Faster-RCNN) in combination with GOTURN architecture [2]. The main work in this paper, the CNN model is used for top view tracking of people in different indoor and outdoor environments. The use of the top view overcomes the various problems encountered in the front view dataset.

The authors briefly explained different tracking algorithms used in literature. They classified them into traditional generic methods, machine learning methods, features, and deep learning-based methods. The authors in the article have tried to explain Faster-RCNN person detection for person detection using an

overhead view video-frames approach. Faster-RCNN has two main steps.

The first step produces region anchors (regions with a probability of occurrence of the probability of occurrence of the object (person)) via the RPN (Region proposal networks). The next step is to classify the object (person) using detected regions and extracts the information from the bounding box [2]. For tracking, the ellipsoid GOTURN is used, which is based on CNN layer architecture.

The authors get this result: the Faster-RCNN detection model achieved the true detection rate ranging from 90% to 93% with a minimum false detection rate of up to 0.5%. The GOTURN tracking algorithm achieved similar results with the success rate ranging from 90% to 94%.

Review on: Long-Term Identity-Aware Multi-Person Tracking for Surveillance Video Summarization

Authors Shoou-I Yu, Yi Yang, Xuanchong Li, and Alexander G. Hauptmann elaborate a study about a multi-person tracking algorithm for very long-term (e.g. month-long) multi-camera surveillance scenarios. The proposed tracker propagates identity information to frames without recognized faces by uncovering the appearance and spatial manifold formed by person detections. The algorithm was tested on a 23-day 15-camera dataset (4,935 hours total).

The authors reviewed work that follows the very popular tracking-by-detection paradigm. They explained carefully the four main components of the tracking-by-detection paradigm object localization, appearance modeling, motion modeling, and data association [2, 3].

The setting was to see the tracking-by-detection-based multi-object as a constrained clustering problem. The location hypothesis that is a person detection result can be viewed as a point in the spatial-temporal space, and the goal is to group the points, so that the points in the same cluster belong to a single trajectory. A trajectory should follow the mutual exclusion constraint and spatial locality constraint, which are defined in the following two constraints:

- Mutual exclusion constraint: a person detection result can only belong to at most one trajectory.
- Spatial-locality constraint: two person detection results belonging to a single trajectory should be reachable with reasonable velocity, that is, a person cannot be in two places at the same time.

The authors propose a tracking algorithm that can be resumed in four main steps: compute Laplacian matrices; compute spatial locality matrix; compute diagonal matrix; compute diagonal matrix. The algorithm was tested in four datasets for experiments terrace1 [8], Caremedia.

> *8h the 15 camera Caremedia 8h dataset is a newly annotated dataset that has 49 individuals performing [3], Caremedia 23d. The 15 camera Caremedia 23d dataset is a newly annotated data set that consists of nursing home recordings spanning over 23 days [3]. The proposed method was compared with three identity-aware tracking baselines* multi-commodity network flow, Lagrangian relaxation, and non-negative discretization. *Therefore, other trackers that did not have the ability to incorporate identity information were not compared [3].*

The findings were able to localize a person 53.2% of the time with 69.8% precision. They further performed video summarization experiments based on their tracking output. Results on 116.25 hours of video showed that they were able to generate a reasonable visual diary for different people, thus potentially opening the door to automatic summarization of the vast amount of surveillance video generated every day.

Review on: Fast Online Object Tracking and Segmentation: A Unifying Approach

To allow online operability and fast speed, the authors adopt the fully convolutional SiamMask framework. Moreover, to illustrate that their approach is gnostic to the specific fully convolutional method used as a starting point, they consider the popular SiamFC and SiamRPN as two representative examples; they then adapt them to propose their own solution to the SiamMask.

The fundamental building block of the tracking system is an offline-trained fully convolutional Siamese network. This compares an exemplar image z against a large search image x to obtain a dense response map. z and x are, respectively, a w by h crop centered on the target object and a larger crop centered on the last estimated position of the target. The two inputs are processed by the same CNN yeilding two feature maps that are cross-correlated. We have in the g_φ equation each spatial element of the response map, which we see as the left side of the equation g_φ referring to the response of a candidate window RoW [9]. For SiamFC, the goal is for the maximum value of the response map to correspond to the target location in the search area x. However, in SiamMask, the authors replace the simple cross correlation with depth-wise cross correlation and produce a multi-channel response map. SiamFC is trained offline on millions of video frames with the logistic laws they refer to as Lsim. The performance of SiamFC was improved by relying on a region proposal network rpn, which estimates the target location with a bounding box of variable aspect ratio SiamRPN outputs box predections in parallel with classification scores, and these are referred as L_{box} and L_{score}. In SiamMask, the authors point out that besides similarity scores and bounding box coordinates, it is possible for the RoW (response of the candidate window) of a fully convolutional Siamese network to also encode the information necessary to produce a pixel-wise binary map. They predict w by h binary masks, one for each RoW using a simple two-layer neural network h_θ. The authors presented two variants; one combines the mask with

rbn parameters box and score, and the other one combines the mask with elsin from cmfc. In order to have the comparison against the tracking benchmarks, it is required to have a bounding box as final representation of the target object. The authors showed that the mbr strategy to obtain a rotated bounding box from a binary mask offers a significant advantage over popular strategies that simply report excess aligned bounding boxes [10–18].

The authors explain that their method aims at the intersection between the tasks of visual tracking and video object segmentation to achieve high practical convenience [11]. However, in addition to tracking the wire bounding box, it also generates the mask and achieves state-of-art performance [12–22]. The performance measure used is the expected average overlap(EAO), which considers both robustness and accuracy of a tracker. As a result, SiamMask can be considered as a strong baseline for OnAVOS. First, it's almost two orders of magnitude faster than accurate approches, and second, it is competitive with recent Video Object Segmentation (VOS) methods that do not employ fine-tuning, while being four times more efficient than the fastest ones; also it doesn't need a mask for initialization.

Review on: A Comparison of Multicamera Person-Tracking Algorithms

Authors A. W. Senior, G. Potamianos, S. Chu, Z. Zhang, and A. Hampapur conducted a study about a comparison of four tracking algorithms that have been applied to people in 3D or 2D multiple cameras in indoor environments. The setting was to present four different approaches that have been taken to tracking a person in indoor scenario instrumented with multiple cameras with overlapping fields of view [1–19]. The first method was the background subtraction tracker. The second tracker uses a radically different approach to the tracking of the speaker, which is the particle filter tracker. The face detection-based tracker was the third method, and the fourth method for tracking the speaker is the edge-based body tracker to use a 3D model-based tracker that they developed for articulated body tracking. The data on which they analyzed tracker performance was collected as part of the CHIL project—a consortium of European Union institutions. All initial data were collected by the University of Karlsruhe in its smart meeting room, and consists of video from four calibrated static cameras mounted in the corners of the 5.9 m by 7.1 m room.

The findings were that in the particle-filtering approach, there is potential for extending this approach to track multiple targets, though occlusion is much more complex, and the feature space is much larger at two dimensions per candidate. The face tracking system relies on face detection, which is not perfect, and cannot be guaranteed with fewer than four cameras, but here works well, and indeed leads to the best system we have hitherto reported on the CHIL data. Finally, the edge alignment technique works very well once initialized but does not recover from tracking failures. The authors suggested a combination, using the particle filtering approach for detection and initialization and the edge alignment approach for tracking may be feasible.

Review on: SimpleTrack: Understanding and Rethinking 3D Multi-Object Tracking

The authors talk about SimpleTrack, which analyzes 3D MOT (Multiple Object Tracking) proposed with some very simple yet effective improvements, and I'm happy to see that many of the improvements are actually adopted by some recent 3D MOT books [27]. Ziqi Pang, Zhichao Li, and Naiyan Wang make the following contributions. First, they summarize a checking by detection framework. Second, they analyze some video cases. Third, they propose some effective solutions, and finally they also rethink some existing benchmarks so that every researcher can compare fairly to each other [16].

To familiarize with 3D MOT visualization, the notion to give 3D MOT is to track objects coherently over time, which includes both localization and also identification. A general tracking by detection framework is we want to associate the detection boundary boxes to the old track list on every frame, and that is to say, we want to use some association matrix to link every detection boundary box to every motion prediction of the tracks. Then we can use the bounty boxes to update the states of the tracks. The first data case we notice is related to how we preprocess the detection boundary boxes, so with the key insight there is a difference between object detection and multiple object tracking, because object detection wants to maximize the map they generally have to output many redundant body boxes just to improve the recall. However, this will confuse the trackers, then we propose to remove the redundant bounding boxes by more regressive nms. Compared to score filtering, this more aggressive nms is even better because it can keep the spatial diversity of the body boxes. The second part we focus on is how we can do better association.

Previously when people do association, they can use some association matrix such as Lu or L2 distance. However if we use IOU, it is not flexible enough if this frame rate is really low, and if the agent is moving abruptly, the IOU may lose target. However, if you lose use of the L2 distance, it is now discriminative enough to be aware of [16]. For example, the orientation and in that case they may associate the response positive detection B to the motion prediction the Blue Bunny boxes instead of the true positive A.

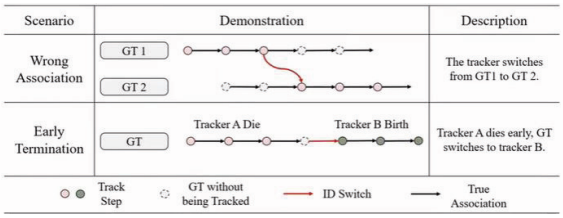

Figure 4. Life cycle management example

To overcome the disadvantages of the two associated matrixes, we propose to use Glu generalized IOU, which combines the best of two worlds, and you can better know both of the methods.

The final part is about how we can do life cycle management. Life cycle management means you know how we can determine if a track is alive or dead. Most of the works they focus on have better association because they think this is the main source of ID switch. However, after doing some data analysis, we found out that the early termination consumes more than 90% of the ID switches, which is really surprising. So, early termination here means that we have a track we initially assign it to, but then we terminate it, and it is switched to idb so that's one ID switch. To avoid this thing from happening, we can use a low-score detection body box to indicate the existence of an object. That is to say, if there is a low-score bounding box corresponding to a track, we don't have to output that bounty box, but we have to keep that track alive. This thing is called a two-stage association, and they really improve the performance.

Finally, if you look at the performance on the web open dataset and nuisance, our method is really competitive compared to some related methods, and this proves that our solutions are simple yet effective [16].

A brief notion about some rethinking of the benchmark we mentioned is that the first is to use higher frame reads, and the second is to use output motion model predictions and design low scores so these two will give you better checking results. Also, it's better for the motor evaluation.

Review on: YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors

This new paper called YOLOv7 the trainable bag of freebies that sets a new state-of-the art for real-time object detectors. As the title suggests, it's a new state-of-the-art model for real-time object detection.

The paper is all about this video introduction. Before getting into this subject, let's understand the history of how people arrived at this paper. Alexi took up the YOLO torch from the original author Joseph Redman, who released the first three YOLO series models. When Redmond quit the computerization industry due to ethical reasons, Alexi maintained his work for the YOLOv3 and also released YOLOv4, wanting to enter the computer vision research stage with cross-stage partial networks that allowed YOLOv4 and v5 to build more efficient features. From that, they discovered YOLOv4 and scaled it, which was the first paper Alexi and Wayne collaborated on. In doing so, they put a YOLOv5 Pytorch implementation over the line. Wang Chiang Yao also released the YOLOr, which introduced new methods with explicit knowledge in neural networks. Now they're joined again to sample something magical, the yolo v7 model. In this article, we are going to talk about the abstract of the paper, how the algorithm works, what approaches they used, why they used the particular methods model comparisons, and finally why is it so awesome. The YOLOv7 trainable bag of freebies sets a new state-of-the-art for real-time object detectors.

The paper states that the model can efficiently predict video inputs ranging from 5 fps to 160 fps. YOLOv7 has the highest average precision of 56.8%. YOLOv7 outperforms both transformer-based object detectors and convolution-based object detectors. Some of the object detectors that YOLOv7 outperform were YOLOr yellow rx, YOLOv5, etc. [17].

The abstract compares it with YOLOv4, and because both of the models are using bag of freebies, the cost of running the model has been reduced by 50% from the same dataset due to its incredible speed and accuracy. The parameters in the hidden layer of neural networks are also reduced up to 40%. Model scaling has never been easy. They can maintain the original model design and structure of a well-performing compound. YOLOv7 has achieved 1.5 times higher average precision than YOLOv4. This is a big deal because YOLOv7 has 75% fewer parameters and 36% less computational time than YOLOv4. How does it work? YOLO uses a sole convolution neural network to predict bounding boxes and class probabilities considering the entire image in a single evaluation in one step. For one unit, YOLO predicts multiple bounding boxes. The class probabilities for each box and all the bounding boxes across the classes make it the one-stage detection model, unlike earlier object detection models, which localize objects and images by using regions of the image with high probabilities of contenders YOLO considers the full image.

Now we'll talk about the architecture of YOLO. Image frames are featured through a backbone, which is then combined and mixed in the neck, and then they are passed along and YOLO predicts the bounding boxes, the classes of the bounding box, and objects of the bounding boxes. Let's understand each of its modules separately. First, the input layer is nothing but the image input you provide. It can be a two-dimensional array with three channels: red, blue, and green. It can also be a video input at each frame of some image input. What is the backbone? It's a deep neural network composed mainly of convolutional layers.

The main objective of the backbone is to extract the essential features. The selection of backbone is a key step, as it will improve the performance of object detection. Oftentimes, pre-trained neural networks are used to train the backbone. Some of the commonly used pre-trained networks are vgg-16 imagenet, routinenet, resnet50, etc.

For YOLOv7, the paper used the following pre-trained weights: vovnet, cspvo. and net Elan. We will learn more about why these weights are used during the study. The object detector models insert additional layers between backbone and head, which are referred to as a copy of the detectors. The essential role of the neck is to collect feature maps from different stages. Usually a neck is composed of several bottom-up parts and several top-down parts for enhancement. We use fpn, rfb, and pan detection happens in the head.

The head is also called a dense prediction to set the director to decouple the object localization and classification task for each module once the detectors make the prediction for this localization and classification at the same time. This layer is present only in one stage after detectors like YOLO ssd rpn into self-detection. They were completely different. Sparse prediction is for two-stage detectors frcnn and rfcn highly different which does the traditional class probabilities for the model input. Our YOLO is one stage. Together they form the YOLO architecture.

Let's dive deeper into the topics and technical words previously mentioned. The first term is bag of freebies model, which refers to increasing the model accuracy by making improvements without actually increasing the training cost. The older versions of YOLOv4 also use the bag of freebies models. In this section, we'll learn some of the trainable bag of freebies used for this particular paper batch. Normalization can be an activation topology, and this part mainly connects the batch normalization layer directly to the convolutional layer. The purpose of this is to integrate the mean and variance of batch normalization into the bias and rate of the convolution layer at the inference stage [17].

Second, implicit knowledge in YOLOr combined with convolutional feature maps in addition and multiplication. Implicit knowledge in YOLOr can be simplified to a vector by pre-computing an inference state. This vector can be combined with the bias and weight of the previous or subsequent convolutional layer. The final ema model is the technique used in mean teacher, and in the system they use the ema model purely as the final interference model training optimizers. The author uses gradient prediction to generate course-defined hierarchical labels. The author also used extended efficient layer aggregation networks and performed a model scaling for concatenation-based models and identifying connections in one convolutional layer [17].

Finally, the author used Microsoft's Coco dataset to train the YOLOv7 from scratch without using any other datasets or pre-trained weights. The model is able to perform with these pre-trained weights only using the Microsoft Coco dataset. During the research, it was figured out that the average precision was higher when the iou threshold was increased. The iou is nothing but intersection over union. It is a term used to describe the extent of overlap of two boxes; the greater the region of overlap the greater the iou. We train a model to output a box that fits perfectly around an object.

For example, in Figure 4, we have a green box and a red box. The green box represents the ground truth, and the red box represents the prediction from our model. The aim of this model would be to keep improving its prediction until the red box and the green box perfectly overlap; that is the iou between the two boxes equals to one coming to layer aggregation networks. The efficiency of a YOLO network's convolutional layers in the backbone is essential to efficient interference speed when started down the

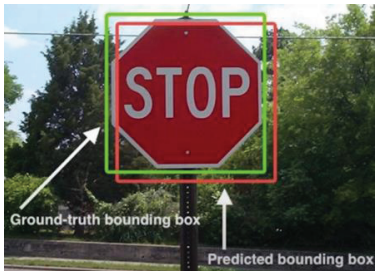

Figure 5. Example model

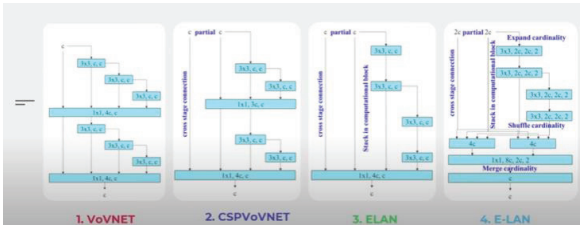

Figure 6. Layer Aggregation Network

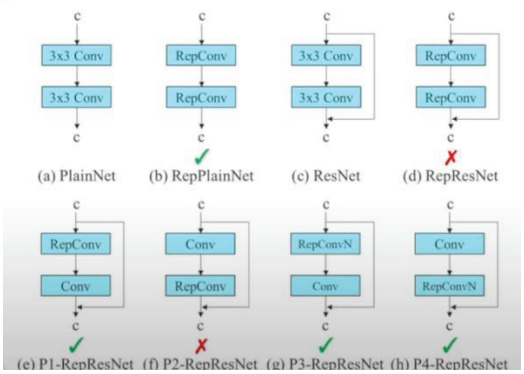

Figure 7. Model re-parameterizing

path of maximum efficiency with cross-state partial networks in YOLOv7.

The authors built and researched what happened to be in this topic, keeping in mind the amount of memory it takes to keep players in memory along with the distance it takes the gradient to back propagate through the layers; the shorter the gradient the more powerful the network will be to learn the final layer aggregation. They chose elan in an extended version of the elan computational block model. Scaling all concatenation-based models will change the input width of some layers and the depth of those models. These provide great support to the model in increasing the accuracy, as the model is now capable of identifying small objects and large objects.

The scaling factors the model is dependent on resolution, depth stage, and width model re-parameterizing, and use gradient flow propagation plans to analyze how re-parameterized convolutions should be combined with different networks. Rep Conv combines three into three convolution, one into one convolution, and identify connections in one convolution layer. Repconv without identity connection is used to design the architecture of planned re-parameterized convolution [17, 26].

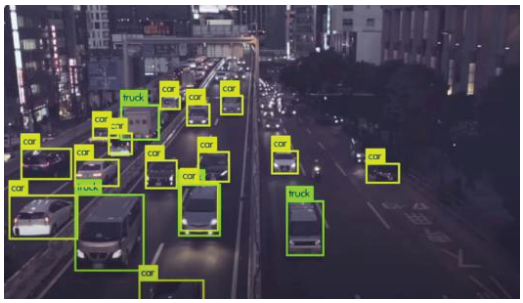

Figure 8. Results

The re-parameterization technique involves averaging a set of model weights to create a model that is more robust to the general pattern that is trying to model in research. There has been a decent focus on model re-parameterization. A piece of the network has its own re-parameterization strategies. The YOLOv7 author uses gradient flow propagation paths to see which model in the network should have re-parameterized strategies, which should not model the level ensemble. The weighted average of the weights of a model of different iteration numbers were used to evaluate this sample module level ensemble training. Multiple identical models with different training data and the average rates of various training model modules level ensemble have been used in YOLOv7 for re-parameterization. We split a module into multiple identical and different module branches during training and integrate multiple branch modules into a completely equivalent module during inference. That module level ensemble is the next topic of the auxiliary head course. To find we call the head responsible for the final output, the lead head and the head used to assist in the training is called the auxiliary head. They use the lead head prediction as guidance to generate a cost of buying hierarchy levels. The reason to do this is that the lead head has a relatively strong learning capability, so soft levels in data from it should be representative of the distribution and correlation between source data and the target. The final level is the same as the soft level generated by the lead head guided label assigner, and the course label is generated by allowing more grids to be treated as a possible target by relaxing the constraints of the possible sample assignment process results. Comparing the model in front of the existing model the algorithm must be impressive and provides a lot of scope for improvement. It is able to predict bounding box properly with high confidence. It's also able to predict images and videos more accurately.

The paper has done an impressive job of presenting this summary. The replacement problem of the re-parameterization model has been overcome by this gradient flow propagation path to analyze how re-parameterization convolutional networks can be combined with different networks. It combines three into three and one into one convolutional. In one convolutional layer, repconvn, the model was able to overcome the problem of dynamic label assignment by using a course to find the lead head guided label assigner. This auxiliary head was really helpful to increase the efficiency of the model paper also introduced extended efficient layer aggregation networks and compound scaling for model scaling [17, 23].

4. Conclusion

Video object tracking is the process of basically monitoring an object throughout a video frame. What that means is you want to localize that object and then you want to be able to predict the trajectory of that object so you know where it's going to be at the next frame. There are different categories of video object tracking. You could do multi-object tracker or single object trackers online or offline—basically, if your models are pre-trained or if it's going on the fly and detection based.

There's a lot of different applications for video object tracking medical imaging and robotics, even in fields like sports analytics. A tracker has two main components. Typically, the detection component has an appearance model and takes advantage so that leverage is spatial features. The detection models work frame by frame, and then you have the object motion model, which will tie those frames together so you can predict where that detection you localized on the individual frame will actually be on the next frame. In the following frames, some of the main challenges in tracking are occlusion are that if you're tracking an object, and either the two objects get very close to each other or if it goes behind some other structure and you lose your actual detections, you'll still have a track on that object because you're project predicting the trajectory, but you won't be able to actually see that object. When you can see it again, you need to be able to accurately pair the track that you had with the trajectory to the detection that represents that specific object. A lot of tracking systems are sensitive to appearance and scale changes, which is one of the main reasons why we use deep learning now in the tracking field because it adds an extra layer that augments pre-existing tracking methods. As you may know, deep learning models are very computationally expensive, so to get those working at a real-time frame rate usually takes 30 frames per second or 60 frames per second,. They have to be pretty lightweight, especially if it's on something like a self-driving car or an aircraft. You can't have so much hardware on it because of the weight, so that's definitely a challenge that needs to be mitigated. I mentioned one of the largest applications just in the commercial sector is self-driving cars. This is a gift from a video from Nvidia actually, and this shows some of the driverless car technology.

That's usually some attract object as assigned a bounding box, and that is also assigned a unique object indicator, which helps you from frame to frame across the video sequence keep track of that what that is and save some information about it. One small difference with this video versus some of the methods that we talked about, the mask method, is it's a segmentation method, so every single picture in the frame has been

identified with the outputs of the systems. Some of the newer detectors like the YOLO algorithm that is tied with a classifier so you'll be able to localize the object and classify it at the same time. With the second component, you have the detection and then you use that detection for your object of interest to instantiate your track. Your track is where you're going to be estimating your trajectory of your object, and the two main ways you can do that are by measurement dynamic models like a common filter [21, 22].

There are deep learning methods that can also be used to predict the trajectory of objects. Typically, computer vision deep learning methods are kind of more in the detector side, so that's a pretty novel advancement, and the specific neural network that is used in this context is an lstm network, a long short-term memory, if anyone is familiar. So once you have your detection object and you have your track, you need to know how they go together. There's an association algorithm component that allows you to calculate the similarity of your detections in your tracks and pair it together so you can update your track estimates with the information that you're going to get from your detection frame. At this point, assuming that the track information is up until the last frame, your detection frame is your current frame, so you update that. You can keep going forward with your predictions and then the output of that is to get your consistent identity label. You want to know that car number one is car number one from the B frame one to the end of the video sequence, and then there's a track maintenance step where basically if you're tracking an object, you don't get detections on it for a certain amount of time. You might want to do something like delete that track or downgrade it's trustworthiness score.

The current state-of-the-art tracking method is not deep learning, and the reason I'm concluding here is because it gives a good understanding of a lot of the components within a tracker and then the deep learning methods build on to it. The current state-of-the-art is called Sort. It's a simple online real-time tracker. This is a multiple object tracker, and it works in real time as stated at the frame rate because it doesn't have a deep learning step. This one is going to be specifically very sensitive to things like occlusions and skill changes. The good thing about this algorithm is it's very lightweight. We can use something for the detection step, which is the first step. We can use something like a CNN-based detection algorithm. It still has to be one of the more lightweight ones, but we can use that with the rest of the algorithm and still have it running at the frame rate. The most commonly used detection algorithm is the YOLO algorithm. The reason why that one is used over some other ones is it stands for "you only look once." How other convolutional neural networks work is there's a sliding window, which basically means you have to pass over the image something like 2,000 times. This only passes over once, and we save a lot of resource consumption. That way, once we have that detection and we instantiate our track based off the detection, we'll have that box from the detection. Then there's the estimation step with a common filter, and that takes in our position state. Our velocity state does some dynamic modeling updates and measurements, and there are different noise models that are included into it as well. That allows us to recursively estimate what our position is going to be on our current frame, so after we have our tracks updated our detections are updated.

Then there's that association step, the common method that's used, which is the Hungarian method, and that's essentially just a cost minimization algorithm that uses the Mahalanobus distance at this metric. That it's minimizing is the reason why that's used is because everything coming out of the Coleman filter is a distribution. So the Mahalanobus distance takes into account the distributions as opposed to something like the Euclidean distance, which is just going to be for a single number. Then there's track maintenance step where you're going to have counters on things like track age and the association history. Information like what detections it's been associated with, how many frames you've seen it for, others you haven't had detections on it, etc. can be used cleverly in order to upgrade or downgrade tracks to get to the object that you want to get to. Some images of A Sort tracker. show that when the target cross over there's that wedge that forms, and that wedge actually indicates that you're searching for detections in that general area. When they cross or there's an occlusion, you lose that information and you have ambiguous detections. So this algorithm is sensitive to that, and the way we fix that sensitivity is with deep learning. The first deep learning algorithm is a deep sort, essentially that sort algorithm.

The main challenge in this topic is to find a balance between computational efficiency and performance. All of these methods have characteristics and limitations under certain circumstances, and they are defined as follows:

Lighting: Light differs in many circumstances; low light adds darkness to the image while higher light adds shadow to the object.

Positioning: Template matching requires a uniform position; otherwise, it cannot detect the object, even if it is present in the image.

Rotation: The image can be rotated in any direction. In this case, some shapes are unable to be identified if the shape matching method is used.

Occlusion: Object behind the object is sometimes not completely visible so it cannot be detected, and the useful part can be ignored [24, 25].

Our goal is to detect and track all objects in a scene, and usually these are the types of scenes that we're looking at. The objects are usually all of the same type, either pedestrian tracking or car tracking, and we have a lot of them so there are a lot of occlusions. There are problems with different viewpoints, and therefore there are different levels of occlusions, depending also on the viewpoint of the camera. We can

also have moving cameras; these are all types of scenes that we want to deal with using a single algorithm.

AUTHORS
Abir Nasry* – Intelligent Processing and Security of System Team, Faculty of Science, Mohammed V University, Rabat, Morocco, e-mail: nasryabir@gmail.com.
Abderrahmane Ezzahout – Intelligent Processing and Security of System Team, Faculty of Science, Mohammed V University, Rabat, Morocco, e-mail: abderrahmane.ezzahout@um5.ac.ma.
Fouzia Omary – Intelligent Processing and Security of System Team, Faculty of Science, Mohammed V University, Rabat, Morocco, e-mail: omary@fsr.ac.ma.
*Corresponding author

References

[1] A. W. Senior, G. Potamianos, S. Chu, Z. Zhang, A. Hampapur. A comparison of multicamera person-tracking algorithms. IBM T. J. Watson Research Center, PO Box 704, Yorktown Heights, NY 10598, USA.

[2] S. Yu, Y. Yang, X. Li, and A. G. Hauptmann. "Long-Term Identity-Aware Multi-Person Tracking for Surveillance Video Summarization," arXiv:1604.07468v2 [cs.CV] 11 Apr 2017.

[3] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. "Multicamera People Tracking with a Probabilistic Occupancy Map," IEEE TPAMI, 2008, the work was supported in part by the Swiss Federal Office for Education and Science and in part by the Indo Swiss Joint Research Programme (ISJRP).

[4] Book Matchmoving: The Invisible Art of Camera Tracking, by Tim Dobbert, Sybex, Feb 2005, ISBN 0-7821-4403-9. Peter Mountney, Danail Stoyanov & Guang-Zhong Yang (2010).

[5] Lyudmila Mihaylova, Paul Brasnett, Nishan Canagarajan, and David Bull. "Object Tracking by Particle Filtering Techniques in Video Sequences," in *Advances and Challenges in Multisensor Data and Information. NATO Security Through Science Series*, 8 (Netherlands: IOS Press, 2007). pp. 260–268.

[6] K. Chandrasekaran (2010). Theses : Parametric & non-parametric background subtraction model with object tracking for VENUS. Rochester Institute of Technology.

[7] L. Bao, B. Wu, and W. Liu. "CNN in MRF: T1: Video Object Segmentation via Inference in a CNN-Based Higher-Order Spatiotemporal MRF," *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, DOI: 10.1109/CVPR.2018.00626.

[8] C. Feichtenhofer, A. Pinz, and A. Zisserman. "Detect to Track and Track to Detect," I*EEE International Conference on Computer Vision*, 2017.

[9] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu. "High Performance Visual Tracking with Siamese Region Proposal Network," *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[10] M. Danelljan, G. Bhat, F. S. Khan, M. Felsberg, et al. "Eco: Efficient Convolution Operators for Tracking," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[11] TY - BOOK, Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip H.S Torr, PY - 2019/06/01 , T1-"Fast Online Object Tracking and Segmentation: A Unifying Approach" , DOI: 10.1109/CVPR.2019.00142ER

[12] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu. "Distractor-Aware Siamese Networks for Visual Object Tracking," *European Conference on Computer Vision*, 2018.

[13] T. Yang, and A. B. Chan. "Learning Dynamic Memory Networks for Object Tracking." In *European Conference on Computer Vision*, 2018. Vol. 1. ISBN 9780549524892.

[14] Background subtraction is the process by which we segment moving regions in image sequences. "Basic Concept and Technical Terms". Ishikawa Watanabe Groupe Laboratory, University of Tokyo. Retrieved 12 February 2015.

[15] JOUR, M., Peter, S. Danail Y, Guang-Zhong. "Three-Dimensional Tissue Deformation Recovery and Tracking: Introducing Techniques Based on Laparoscopic or Endoscopic Images," JO-*IEEE Signal Processing Magazine*, 27, July, 2010.SP-14, EP-24.

[16] Z. Pang, Z. Li, and N. Wang. "Simpletrack: Understanding and Rethinking 3D Multi-Object Tracking," arXiv:2111.09621v1 [cs.CV] 18 Nov 2021.

[17] C. Wang, A. Bochkovskiy, and H. M. Liao. "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," arXiv:2207.02696v1 [cs.CV] 6 Jul 2022.

[18] P. Dai, R. Weng, W. Choi, C. Zhang, Z. He, and W. Ding: "Learning a Proposal Classifier for Multiple Object Tracking." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, 2443–2452.

[19] TY - BOOK, J. N. Zaech, A. Liniger, D. Dai, M. Danelljan, and L. Van Gool. "Learnable Online Graph Representations for 3D Multi-Object Tracking," *IEEE Robotics and Automation Letters*, 2022 PY - 2021/04/23.

[20] L. Lin, H. Fan, Y. Xu, and H. Ling. "Swintrack: A Simple and Strong Baseline for Transformer Tracking," arXiv preprint arXiv:2112.00995, 2021.

[21] J. Pang, L. Qiu, X. Li, H. Chen, Q. Li, T. Darrell, and F. Yu: "Quasidense Similarity Learning for Multiple Object Tracking," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 164–173.

[22] F. Zeng, B. Dong, T. Wang, X. Zhang, and Y. Wei. "Motr: End-to-End Multiple-Object Tracking with Transformer," arXiv preprint arXiv:2105.03247, 2021.

[23] J.-N. Zaech, A. Liniger, M. Danelljan, D. Dai, and L. Van Gool. "Adiabatic Quantum Computing for Multi Object Tracking," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8811–8822.

[24] X. Han, Q. You, C. Wang, Z. Zhang, P. Chu, H. Hu, J. Wang, and Z. Liu. "Mmptrack: Large-Scale Densely Annotated Multi-Camera Multiple People Tracking Benchmark," arXiv preprint arXiv:2111.15157, 2021.

[25] X. Zhang, X. Wang, and C. Gu. "Online Multi-Object Tracking with Pedestrian Re-Identification and Occlusion Processing," *The Visual Computer*, vol. 37, no. 5, 2021, pp. 1089–1099.

[26] K. Cho, and D. Cho. "Autonomous Driving Assistance with Dynamic Objects using Traffic Surveillance Cameras," *Applied Sciences*, vol. 12, no. 12, 2022, p. 6247.

[27] A. Cioppa, S. Giancola, A. Deliege, L. Kang, X. Zhou, Z. Cheng, B. Ghanem, and M. Van Droogenbroeck. "Soccernet-Tracking: Multiple Object Tracking Dataset and Benchmark in Soccer Videos," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3491–3502.