Titre: Title:	Real-Time Vehicle Detection and Tracking From Surveillance Cameras in Urban Scenes
Auteur: Author:	Oumayma Messoussi
Date:	2021
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Messoussi, O. (2021). Real-Time Vehicle Detection and Tracking From Surveillance Cameras in Urban Scenes [Master's thesis, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/6657/

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Real-Time Vehicle Detection and Tracking From Surveillance Cameras in Urban Scenes

OUMAYMA MESSOUSSI
Département de génie informatique et génie logiciel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie informatique

Juin 2021

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Real-Time Vehicle Detection and Tracking From Surveillance Cameras in Urban Scenes

présenté par **Oumayma MESSOUSSI**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Michel DESMARAIS, président
Gabriela NICOLESCU, membre et directrice de recherche
Guillaume-Alexandre BILODEAU, membre et codirecteur de recherche
Sarath Chandar ANBIL PARTHIPAN, membre

DEDICATION

To my dearest father and mother,
the best in the world.
I hope I make you proud . . .

ACKNOWLEDGEMENTS

First, I would like to thank my research supervisors, Prof. Gabriela Nicolescu and Prof. Guillaume-Alexandre Bilodeau, for giving me the invaluable opportunity to do research under their guidance. It is their continuous support that made the past two years fruitful and enlightening.

I also sincerely thank the entire team at Cysca Technologies, for providing me with the opportunity to do research and closely merge it in a professional context.

I extend my gratitude to Mitacs for providing the funding for this work.

I also want to express my heartfelt thanks to my colleagues at the HES lab for the pleasure of their company.

I sincerely thank the members of the jury for taking time to review my thesis and to give valuable feedback.

Finally, I want to thank my family and friends all over the world. They have always been there for me. It is their support that enables me to go further and further.

RÉSUMÉ

En vision par ordinateur, la manipulation de vidéos de caméras de surveillance pour extraire des informations sur les usagers de la route est un domaine de recherche très actif et bien étudié. Dans le cadre de ce memoire, nous nous intéressons à la détection et le suivi des véhicules en milieux urbains au moyen d'une approche de suivi par détection. Nous nous sommes spécifiquement concentrés sur la mise en œuvre d'une méthode qui effectue les tâches ci-dessus en temps réel tout en améliorant les performances, en particulier dans des cas de grands déplacements spatiaux et/ou des occlusions à long terme.

Composée de deux étapes, notre méthode génère les détections à l'aide du modèle SpotNet pour localiser les véhicules dans des trames vidéo et fournir des informations telles que leurs boîtes englobantes, la catégorie de véhicule et le score de confiance. Ensuite, nous étendons une méthode de suivi multi-objets basée sur la métrique IOU et le suivi visuel pour les occlusions à court terme, appelée V-IOU, avec des caractéristiques de ré-identification pour effectuer l'association des données et construire les trajectoires finales. A chaque trame, la détection qui atteint le score IOU le plus élevé avec une trajectoire existante est ajoutée à cette derniere. Lorsqu'une trajectoire n'est pas reliée à une nouvelle détection, une méthode de suivi visuel est utilisée pour prédire les emplacements suivants de l'objet, compensant ainsi les occlusions à court terme. Nous proposons de combiner le score IOU avec la similarité cosinus entre les vecteurs de ré-identification de la paire détection/trajectoire. Les vecteurs de ré-identification décrivent l'apparence du véhicule à travers des images prises à différents angles et/ou moments, ce qui nous permet de faire correspondre les véhicules dans des scénarios avec de plus longues périodes d'occlusions et/ou un grand déplacement spatial dû à une vitesse relativement élevée, qui donnent des boîtes englobantes éloignées.

Cette méthode proposée permet d'améliorer les performances sur le benchmark de UA-DETRAC, notamment en termes de précision de suivi et de taux de fausses alarmes, et surtout d'atteindre des vitesses de traitement en temps réel allant jusqu'à 60 trames par seconde. L'étude d'ablation valide l'impact des caractéristiques de ré-identification en termes de métriques PR-MOTA et PR-MOTP. Dans les travaux futurs, nous pouvons tester l'impact de l'utilisation de la metrique IOU sur les masques sur l'association de données pour tirer parti des segmentations sémantiques générées par SpotNet. Nous pouvons également incorporer une autre information simple dans l'étape d'association qui est le type de véhicule.

ABSTRACT

In computer vision, mining videos from surveillance cameras for information about road users is a very active and well-studied area of research. This work focuses on vehicle detection and tracking in urban settings using a tracking-by-detection approach. We specifically focused on implementing a method that performs the above tasks in real-time while improving performance, especially with large spatial displacements and/or long-term occlusions.

Composed of two steps, our method generates candidate detections using the SpotNet model to locate the vehicles in video frames and to provide information such as their bounding boxes, vehicle category and confidence score. Next, we extend a multi-object tracking method based on the IOU metric and visual object tracking for short-term occlusions, referred to as V-IOU, with re-identification features to perform data association and construct the final trajectories. At each frame, the detection that achieves the highest IOU score with an existing track is added to this trajectory. When a track is not linked to a new detection, a visual tracking method is used to predict the next few locations of the object, thus compensating for short-term occlusions. We propose to combine the IOU score with the cosine similarity between the re-identification feature vectors of the detection/track pair. The re-identification cues describe the vehicle appearance through images taken at different angles and/or times, hence allow us to match the vehicles in scenarios with longer periods of occlusions and/or a large spatial shift due relatively high speed that create distant bounding boxes.

This proposed method leads to improved performance on the UA-DETRAC benchmark, particularly in terms of tracking precision and false alarm rates, and most importantly achieves real-time processing speeds of up to 60 frames per second. The ablation study validates the impact of the re-identification features in terms of PR-MOTA and PR-MOTP metrics. In follow-up works, we can test the impact of using masked IOU on the data association to take advantage of the semantic segmentations generated by SpotNet. We can also incorporate another simple information into the association step like the vehicle category.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

AI	Artificial Intelligence
MOT	Multi-Object Tracking
CNN	Convolutional Neural Network
IOU	Intersection Over Union
FPS	Frames Per Second
PR	Precision-Recall
AP	Average Precision
ReID	Re-IDentification
TTL	Time To Live

CHAPTER 1 INTRODUCTION

This first chapter discusses the motivation behind our work and the advantages of vehicle detection and tracking systems in modern applications, then introduces the elements of our problematic, our research objectives and contributions.

1.1 Context and motivation

Computer vision is the field of Artificial Intelligence (AI) that aims to teach computers how to "see" like humans, i.e. how to process visual data and derive meaningful insights from it [18]. This allows machines to describe objects, distinguish them from one another, understand how they move, etc.

In the past few years, the amounts of visual data available have increased tremendously due to the large number of smart devices, cameras and sensors everywhere. There is a good chance that the person reading this thesis has a smartphone with at least two cameras. according to [19], the number of connected devices world-wide would reach 46 billion in 2021, i.e. nearly 6 devices per person. This results in about one million video minutes on the internet every second [20]. So, it became impossible to rely on human manual efforts to understand and mine these videos and images for information, and greatly important to develop algorithms for machines to perceive the world. Various computer vision techniques combined with the power of deep learning can handle such data elegantly [21] and allow us to describe objects of interest and how they move in a video.

Computer vision is a field applied to many areas from healthcare all the way to engineering and transportation. In modern traffic monitoring infrastructures, surveillance cameras became widely adopted and easily accessible, making them a staple in these systems. It was estimated that there would be about 45 billion cameras around the world by 2022 [22]. These cameras generate large volumes of data that are relevant to many transportation-related applications such as parking lot management, vehicle detection and recognition, vehicle and pedestrian counting, and license plate recognition. With a tendency towards creating more smart cities, these applications allow for more efficient transportation planning and traffic flow optimization [23], and overall improved road user safety.

1.2 Problematic and definitions

1.2.1 Problem statement

Intelligent transportation systems that rely on cameras among other sensors, play an important role in improving efficiency and safety in smart cities. As an example, autonomous vehicles require advanced computer vision techniques to perceive their environment and know how to interact with the surrounding objects. They need to be able to identify road signs, traffic lights, pedestrians and vehicles, in order to be able to navigate their way between two points. Other examples include detecting road infractions and managing parking spaces. The latter requires detecting when vehicles enter the parking area and tracking their movement all the way until reaching the parking spot. This helps automate the parking management avoiding, thus, manual labor and reducing costs. Describing a vehicle trajectory is also critical for the user's safety since it traces their movement and consequently can show how it interacts with nearby moving objects to avoid collisions. So, video surveillance bares numerous benefits both for the road user and the entities managing them.

In video surveillance, objects of interest such as vehicles are localized then tracked throughout the video frames. This way, using multi-target detection and tracking techniques, a surveillance system can provide information about an object category, location and trajectory.

Multi-object tracking is a very active area of research in computer vision, mainly due to the multitude of challenges it comes with. Similar to when a human struggles to keep track of several objects in a video, a tracking model also faces difficulties due to likely intersecting tracks which can alter identities or create gaps in tracks. Below are some of the main difficulties observed:

- **Occlusion**: The view of a target can be partially or fully occluded by another target or a background object. This causes the object features to become inaccurate or outdated during the occlusion period. With partial occlusion, the features of the forefront target interfere with those of the occluded target. Full occlusion, on the other hand, makes the object features irretrievable. These situations can lead to mistakenly switching the identities of the two targets, temporarily losing the target which fragments the trajectory, or even early termination of a track.
- **Fast motion**: When a target is moving too fast in a video, its location changes significantly between consecutive frames. This makes the spatial information about a target unreliable. Fast movement can also lead to motion blur which makes appearance descriptors inaccurate.

- **Scale changes**: When an object moves closer or further away from the camera, its scale changes accordingly, and consequently its size would increase or decrease as well. Therefore, the tracking method needs to adapt to these changes and update the target bounding box to avoid losing information about the full object or introducing background information by error.

- **Illumination variations**: Changes in lighting around the targets, due to weather for example, can alter objects appearance cues. When an object is in direct exposure to sunlight, this causes light reflection on vehicle surfaces which may alter the color of an object. A cloudy sky or a night scene also greatly change what an object looks like. For example, the headlights of a vehicle in the dark act like spotlights on the camera which prevents it from capturing the object correctly. Consequently, these issues make tracking more difficult and prone to identity switches.

- **Target similarity**: Tracking multiple objects of the same category increases the difficulty of the task because of the high similarity in terms of appearance. This makes it hard to distinguish between two or more targets and correctly maintain their distinct identities.

Thus, within this project we are interested in developing methods to detect and track multiple vehicles in single-camera views as accurately as possible and in real-time speeds. We are also particularly interested in addressing the problems of motion-related displacement and long-term occlusion. There are many ways to approach this problem which we will review in the next chapter. Particularly, we will tackle this problem from a tracking-by-detection perspective which performs detection and tracking as two consecutive steps. The first task is referred to as multi-object detection and it localizes the vehicles in the images. The second step is multi-object tracking which assigns unique identifiers to the detected objects, extracts information describing them and links detections in each frame to establish trajectories. An example of what the output of our system looks like is shown in figure 1.1

More specifically, the two-step pipeline accomplishes the following tasks:

- Detect vehicles in video frames and provide the pixel coordinates of bounding boxes surrounding each vehicle, as well as confidence scores and vehicle type.

- Track the vehicles thought the frames by linking detections to tracks using spatial information from the bounding boxes and appearance features that encode the object color, shape, size, etc.

- Perform detection and tracking in real-time by ensuring a choice of models and techniques that yield a good trade-off between precision and inference time.

Figure 1.1 Sample frames showing the output of our detection and tracking pipeline

1.2.2 Definitions

We describe below the two key concepts invoked in our work and define the computer vision tasks related to them.

Vehicle detection

In computer vision, the task of object detection deals with identifying the locations of objects of interest in an image. Unlike in object segmentation or semantic segmentation which aim to create homogeneous groups of pixels matching certain criteria and/or semantic value [24], object detection often localizes objects by giving coordinates of a box around the object.

The aim is to identify objects in an image quite similarly to how a human would see them. For any image, there many possible ways to define a bounding box around an object. For example, when a vehicle is partially occluded, a bounding box could be defined as surrounding only the visible part of the vehicle, whereas most datasets would define a box that outlines the entire vehicle including the hidden parts.

We are interested in using a model that is able to see and identify vehicles in urban scenes with various challenges like occlusions and weather changes. Such challenges make object detection a very challenging task that requires continuous research and the use of appropriate datasets tailored to the application. Our work supports the idea that robust object detection that allows to effectively localize objects in challenging scenes is crucial for the overall performance of the object tracking method.

Single-camera vehicle tracking

The second part of our system aims to provide the trajectories, often called tracks, describing the movement of vehicles in the videos. Unlike visual object tracking which maintains the identity of only one object [25], multi-object tracking locates more than one object, each with a distinct identifier, in subsequent video frames.

To do so, at each frame, new candidate detections are matched with established tracks from previous frames using certain association criteria and characteristics of the objects. Detections that do not match with any existing tracks initialize new tracks, and tracks that do not match with any detections for a number of frames will be terminated. Therefore, tracks are sequences of detections belonging to the same object in a video. Each track is given a unique identification number which should be maintained.

The process of linking detection candidates to tracks is often referred to as data association. This process requires the use of feature descriptors that encode information about the objects. Once the features are extracted, we then apply an affinity measure to compute the association cost for each detection-track pair. Lastly, based on the computed costs, final optimal associations are selected and tracks are, then, established. Figure 1.2 illustrates a typical tracking-by-detection workflow.

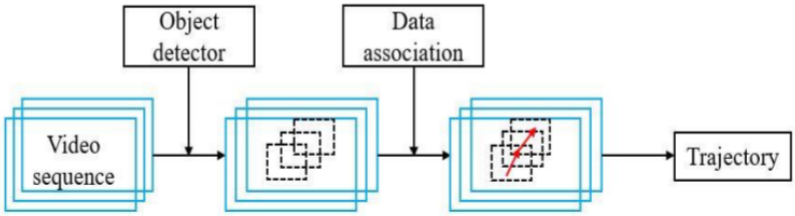


Figure 1.2 Steps of a typical tracking-by-detection MOT method [4] (© 2019, IEEE).

1.3 Scope of the project

This work aims to develop a system able to process video frames from surveillance cameras in urban settings such as parking lots and roads. The objects of interest are vehicles from various categories like cars, buses, vans and trucks. Videos recorded from these scenes present a wide array of challenges such as illumination changes and noise due to weather conditions, partial/full occlusions and different vehicle speeds.

Throughout this project, we assume as input videos from fixed cameras installed in an urban

setting above ground level, and a fixed resolution for the videos. For all our training and tests, we use a dataset that presents the above mentioned challenges.

Our focus, during this project, was on selecting a proper object detection model to provide vehicles locations, and developing a tracking method to construct their trajectories. Therefore, we deal with two fundamental computer vision tasks: multi-object detection and multi-object tracking. With the rise of deep learning techniques in the past few years, these tasks have seen significant improvement.

Our system is subject to the following constraints:

- Real-time processing: end-to-end video frame processing, including detection, feature extraction and association, must be performed in real-time or near real-time (more than one frame per second).
- Robustness against challenging scenarios: our system should ensure solid performance across challenging scenes with various levels of occlusion, motion/speed and weather-related challenges.

This work has been conducted in collaboration with an industrial partner, Cysca Technologies, that specializes in the development of integrated systems engineering solutions. Therefore, we also focused on system efficiency in terms balance between competitive performance metrics such accuracy and precision, and total processing time to ensure overall system timely responsiveness. Our work was also introduced as part of a comprehensive platform for vehicle tracking and geo-localization in a multi-camera environment. By performing previously presented vehicle detection and tracking in each camera view, the next phase is applying multi-camera re-identification to match tracks, creating unified coherent trajectories for each vehicle. The multi-camera-focused research was conducted as part of another master's thesis.

1.4 Research objectives and contributions

The goal of this work is to develop an efficient vehicle detection and tracking system that processes videos from a surveillance camera in real-time, while also being robust against a variety of challenges in urban scenes. Hence, we aim at achieving a healthy trade-off between performance in terms of task-related metrics and processing speed. We can summarize our research objectives as the following:

- Select an appropriate object detection model for vehicles in urban scenes.

- Develop a tracking method to handle long-term occlusions and large displacements due to fast motion by combining spatial and appearance features for data association.
- Validate our proposed method on a public dataset/benchmark as well as videos from Cysca.

With the previous objectives in mind, our contributions in this thesis are:

- Performance improvement in terms of multi-object tracking metrics by extending data association with re-identification features.
- Real-time end-to-end detection and tracking.

1.5 Thesis outline

This thesis is comprised of the following chapters: Chapter 2 presents previous work on the mains tasks of our project. Then, in chapter 3, we introduce the approaches and methods we used. Chapter 4 dives in detail into the implemented method and achieved results. Finally, we end with conclusions and future works in chapter 5.

CHAPTER 2 LITERATURE REVIEW

This chapter introduces some basic concepts as well as state-of-the-art approaches in each of the two components of our work, detection and tracking.

2.1 Multi-object tracking

Before reviewing previous Multi-Object Tracking (MOT) works, we start by summarizing some of the most used feature descriptors for MOT during the last few years:

`Spatial features:`	`represent an object location in the image` example: intersection over union (IOU) between bounding boxes, distance between object centers, scale.
`Motion features:`	`describe location displacement/movement` optical flow, deep motion features.
`Appearance features:`	`encoding the visual aspect of an object` color histogram, deep appearance features from convolutional neural networks (CNNs), re-identification (ReID) features.
`Class cost:`	`associating a class cost by comparing class labels between detections and tracks.`

Solving computer vision problems has long been a problem of how to represent features in an image, which encode characteristic information required for a particular task.

As briefly presented in the introduction, MOT is a widely researched topic in computer vision due to its various applications but also to the yet unresolved challenges it faces. In recent years, object detection methods have achieved remarkable performances thanks to research in deep learning and advanced architectures of CNNs [9, 10, 12, 13, 26]. Therefore, the majority of top-performing MOT methods use the tracking-by-detection scheme to take advantage of the strong detection results. This approach applies an object detector at each frame, then the focus of the MOT method is on yielding the best association between objects.

Unlike batch tracking approaches like IHTLS [27], revisited JPDA [28] and GOG [29] which process a number of frames or a full video sequence before yielding results, online MOT methods process one frame at a time without having visibility or information about future frames. Therefore, online MOT methods are naturally more suitable for real-time systems, although not all online MOT algorithms perform in real-time. Batch, or offline, MOT methods inherently perform better in terms of MOT performance metrics due to accessibility to information from past and future frames. For example, such information allows offline methods to link/fill gaps in fragmented tracks (when the trajectory of an object is "cut off" into different segments with different identifiers). Hence, online MOT is a more challenging task by nature.

2.1.1 Tracking by detection

Representative tracking-by-detection MOT methods are composed of 4 key units: detection, object state propagation, data association and finally track lifespan management. Earlier online tracking works like SORT [30] propose a simple method focused on improving frame-to-frame association. To do so, they use Faster R-CNN [9] as the object detection model and each object state is described by:

$$x = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T, \tag{2.1}$$

where (u, v) is coordinate pair of the center pixel of the bounding box, s and r are respectively the scale and aspect ratio. The velocity and aspect ratio are assumed constant, so when no detection is linked to a track, the state is updated by the prediction using a linear constant velocity model. When a detection is associated, the state is updated with the vector x using the detection information and a Kalman filter [31] motion model to solve the velocity components. For the data association step, the intersection-over-union (IOU) metric is used to measure the amount of overlap between each pair of candidate bounding boxes and a track predicted box. This way, a cost matrix is computed and the Hungarian algorithm [32] along with a minimum overlap threshold are applied to solve the assignment. Lastly, tracks that have not been matched with any detections for a defined number of frames (set to 1) are terminated.

While this method is able to process video sequences at a speed of up to 260 frames per second (FPS) as reported, it solely takes into consideration the spatial information regarding detections for association. This makes the method struggle with long-term occlusions which increases the number of ID switches.

As an improvement to this method, DeepSORT [33] was later proposed. While keeping some of the same basic elements like the Kalman filter and Faster R-CNN detections, it adds a CNN that has been trained on a person ReID dataset in order to extract deep features describing a query object. These deep features are used as appearance information along with the IOU score to assign detections to tracks between frames. The data association step is, hence, tweaked to compute a weighted sum of the Mahalanobis distance between predicted and measured states, and the smallest cosine distance between the appearance descriptor of the current detection and the last 100 descriptors in a track.

Another work, similarly to SORT, aimed to simplify the MOT task even further, basing their approach exclusively on the IOU metric. As illustrated in figure 2.1, the IOU tracker [5] strips down the data association component to a simple IOU between detected bounding boxes in frames pairs. The highest IOU score between a track last registered bounding box and a new detection is selected if above a configurable threshold. Unmatched detections from the current frames initialize new tracks, and unmatched tracks are discontinued after a number of frames.

Without resorting to visual or motion features, the authors defend the idea that having strong high-accuracy detections allows for using simpler tracking methods to achieve higher processing speeds whiles still competing with more complex MOT methods in terms of tracking performance.

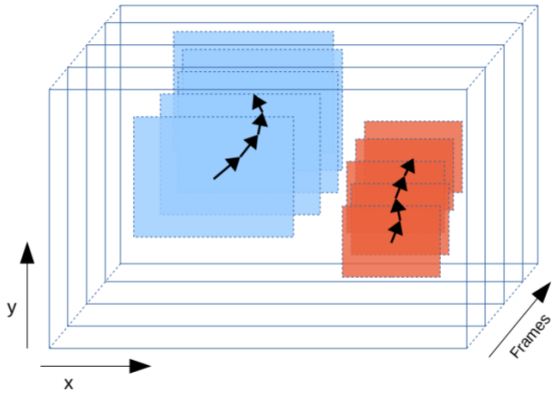


Figure 2.1 Principle of the IOU tracker: linking detections between frame using IOU [5] (© 2017, IEEE).

However, when the detector fails to consistently detect an object, a small number of missed detections can greatly increase the number of ID switches and fragmentation. That is why the IOU tracker was, later, revisited. The extended V-IOU tracker [2] introduces the use of a visual object tracker to predict a target location when detections are not matched, as

shown in figure 2.2. In this case, when a track is not updated by one of the current frame detections, a visual object tracking method is called to predict the object next locations for a fixed number frames, referred to as TTL (time to live), which is decremented with each prediction. If after TTL frames no detection is matched with the track, then it is terminated. Else, TTL is restored to initial value and track is continued. Backward visual tracking is also performed when new tracks are instantiated to potentially link them with previously finished tracks.

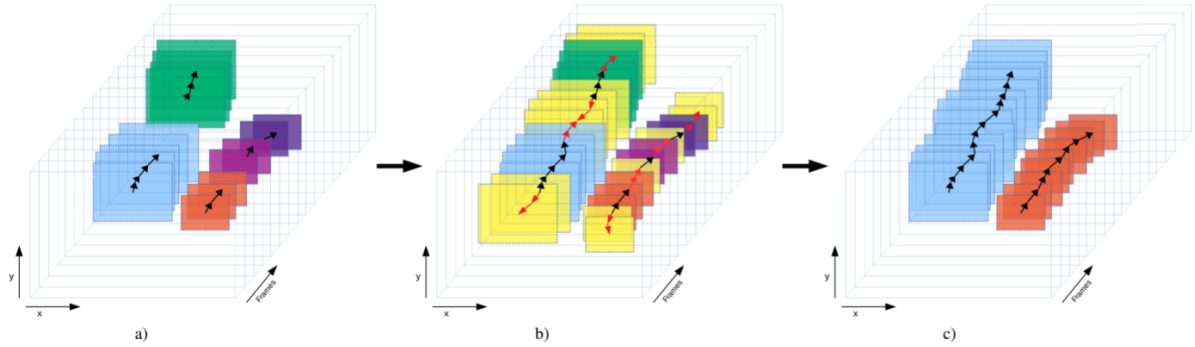


Figure 2.2 Principle of the V-IOU tracker: (a) IOU tracking results are prone to fragmentation. (b) Visual object tracking can fill the gaps (yellow portions) when detections are missing. (c) End results with linked consistent tracks [2] (© 2018, IEEE).

By combining IOU with visual object tracking, V-IOU tracker maintains a good balance between speed and tracking performance, with reported inference speed of over 200 FPS which is its main selling point. Yet, there is still room for improvement as this method does not incorporate appearance features for full occlusion scenarios where the visual tracker fails. This can potentially improve performance with long-term occlusions and/or reduce identity switches. The IOU cost for association is also vulnerable with high speed. When an object moves fast between two frames, the detected bounding boxes can be far away from each other causing very little or no overlap. This pushes the IOU score towards zero which prevents the boxes from being linked even if they represent the same object.

Most recently, a work conducted by Kang et al. and presented in a master's thesis [3] builds on the IOU tracker and adds a Kalman filter to predict an object next location when the IOU theshold is not met. If the prediction also fails to meet a minimum spatial overlap, then then method uses appearance features from a vehicle ReID model to perform the association. While this approach achieves results comparable to the state-of-the-art on the UA-DETRAC challenge, we argue that using the ReID features can be a great asset for data association both when matching new detections to tracks, and when matching predicted locations to

detections as well.

The previous methods modeled data association as an optimization problem, while other works approach it with end-to-end neural networks, such as DAN [34] and DMAN [35].

DAN, short for Deep Affinity Network, automatically learns objects appearance features and their affinities at several abstraction levels. It computes the correlation between objects using pairs of extracted feature vectors. This way, it can link objects in multiple frames by using the computed affinity along with the Hungarian algorithm. However, the reported speed of this method is quite low at about 6 frames per second.

Dual Matching Attention Networks (DMAN) proposes the use of a single object tracker with a cost-sensitive loss function, as well as temporal and spatial attentions. Given detections in a frame, visual object tracking is performed to track each target. When, tracking results deteriorate, tracking is stopped and data association is performed to compute similarity with unmatched detections. The cost-sensitive tracking loss models the attention part by instructing the model to focus on hard negative distractors. Spatial attention during data association helps the model to focus on matching patterns in pairs of images, whereas temporal attention reduces noise in observations by assigning various levels of attention to different samples. While DMAN, achieves comparable performance to the top two methods on the MOT17 challenge [36] that year, its reported speed of 0.3 frames per second is very low for many real-world applications.

Other works make use of recurrent neural networks as a way to model temporal relations between targets. A proof-of-concept joint detection and MOT method was presented in [6] that uses a single neural network to detect and track objects. A recurrent neural network is employed to combine detections from a SSD [37] model into tracks as shown in figure 2.3. The association is based on spatial cues, appearance features extracted from the detector, as well as detection and tracking scores. The initial results presented show that this method scores high values of false positive and ID switches, therefore the authors suggested the use of ReID for occluded objects as well as swapping SSD for a stronger detector.

Similarly, Milan et al [38]. proposed a recurrent neural network-based tracking method that is end-to-end trained. Similarly to the previous method, it links detections to tracks using the recurrent network but no appearance features are used.

A graph-based approach was proposed by Braso et al. which exploits the typical network flow of MOT. The neural solver for MOT [39] method learns to predict final trajectories directly from graphs instead of computing pairwise costs then using a solver. To do so, learning is performed on the MOT graph domain with a message passing network (MPN) that uses

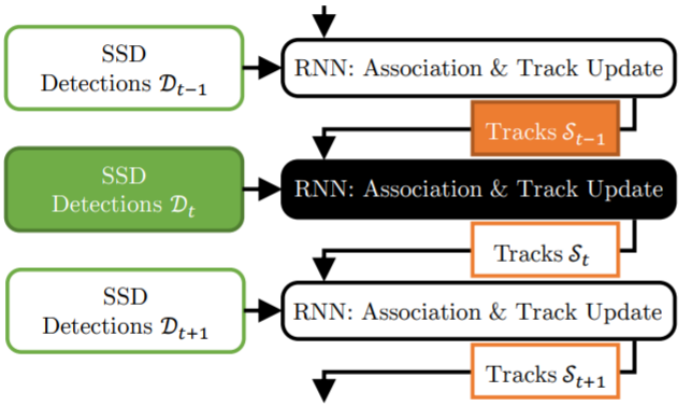


Figure 2.3 Illustration of joint detection and multi-object tracking principle: Detections from each frame are used to update the object state and perform data association at the RNN level [6] (© 2018, IEEE).

appearance and geometry features. Thus, this method can reason globally over of detections and predict final tracks. However, graph-based methods are expensive to optimize which limits their usability in real-world applications.

Leal-Taixe et al. present an appearance-driven 2-stage method referred to as Learning by tracking [7]. The first component is a Siamese CNN to estimate likelihood that two image patches should be linked to the same trajectory. To represent the objects, pixel values and optical flow information are combined. Another set of contextual features is constructed from the position and size of detections. The latter are merged with the CNN features using a gradient boosted classifier yields a matching probability, and using linear optimization, final tracks are constructed.

Lee et al. proposed an improvement referred to as FPSN [40], for Feature Pyramid Siamese Network, which used a Feature Pyramid Network to extract target features from various levels, thus constructing a multi-level discriminative feature. While typical Siamese networks lack motion information, FPSN-MOT adds spatio-temporal motion features to overcome this issue.

Making use of pixel-level information, Osep et al. presented a tracking by segmentation method called Track, then Decide: Category-Agnostic Vision-based Multi-Object Tracking [41]. This method is model-free and uses category-agnostic image segmentation to generate object segmentations. Objects are then tracked by performing mask-based association on pixels, hence utilizing semantic information.

In the same dynamic, Track R-CNN [42] was proposed which, first, annotates two track-

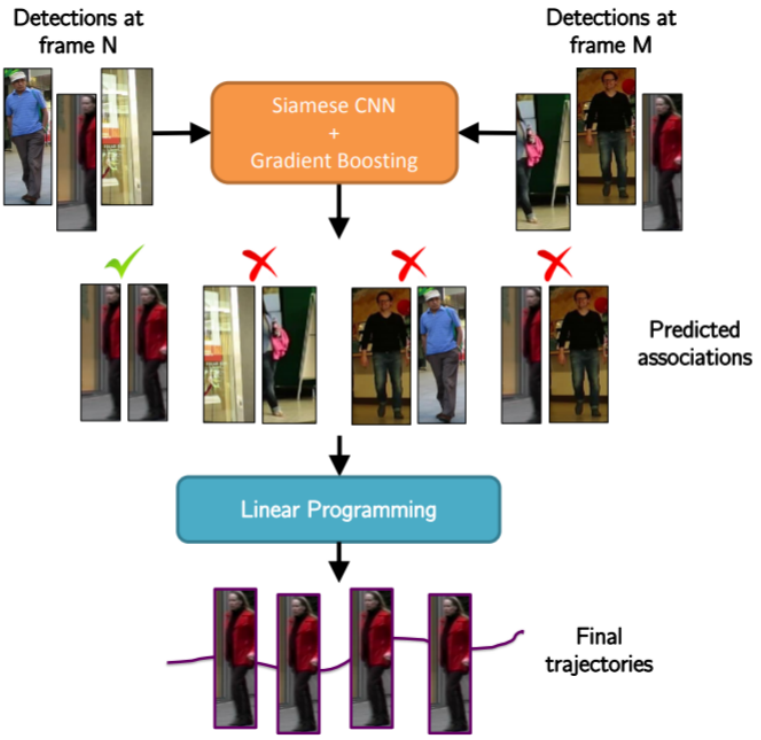


Figure 2.4 Architecture of Learning by tracking method for MOT: a predicted matching probability is estimated by a Siamese CNN combined with gradient boosting, before using a linear optimization to return final tracks. [7] (© 2016, IEEE).

ing datasets with segmentations using a semi-automatic approach. Mask R-CNN detection model is extended with 3D convolutions. The work also includes the proposal of a new metric called sMOTSA, soft MOT and segmentation accuracy, to evaluate the performance of segmentation, detection and tracking jointly. Hence, this work proposed a full environment comprised of new datasets, a MOT method and evaluation metrics

2.1.2 One-shot detection and tracking

Another promising perspective to MOT is joint detection and tracking, or single-shot detection and tracking, which implies the use of a single network. As presented in [8], using a one-shot detector referred to as Joint Detection and Embedding (JDE) model, it learns the detections and appearance vectors in the same network. As shown in figure 2.5, they used a Feature Pyramid Network (FPN) to allow predictions to be made from different scales. For the association, they used the cosine distance between appearance embeddings, the Mahalanobis distance as motion affinity and finally the Hungarian algorithm to solve assignment. This method mainly reduces computational costs and performs in near real-time.

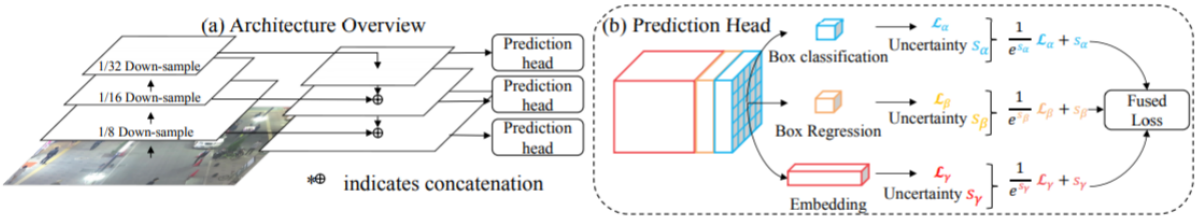


Figure 2.5 Architecture of the JDE framework: (a) the FPN network architecture and (b) the prediction head. The learning of JDE is designed as a multi-task learning problem [8] (© 2020, Springer Nature Switzerland AG).

A similar work proposes a method for tracking "without bells and whistles" [43] , referring to a tracker that does not target the key elements such as motion prediction or occlusion handling. Instead, their Tracktor performs MOT using solely an object detection model adapted to predict the location of objects in the next frame using the already existing regression head. No explicit association is computed but instead, a Siamese network is used for ReID as well as camera motion compensation. Thus, this method performs tracking without training or optimization on tracking-oriented data.

Zhang et al. tackle the scalability problem in end-to-end deep neural networks: MOT methods that use object features to estimate target motion and carry-out pair-wise ReID struggle as the number of targets grows. They propose FFT [44], or Flow-Fuse Tracker, which introduces two techniques to solve the issue. The first is target flowing using a deep neural network model called FlowTracker that learns an indefinite number of target-wise motion patterns from optical flow. The second technique is target fusing where another deep neural network model, FuseTracker, refines candidate detection generated from FlowTracker and an object detection method, and fuses them.

Zhou et al. proposed tracking objects as points [45] which simultaneously performs detection and tracking. The model called CenterTrack applied a detector to an image pair in addition to detections from the previous frame. Targets are represented by the center of their bounding boxes, which is yielded by the detector CenterNet. The detector is also trained to regress an offset between an object current center and its center in the last frame. Data association is exclusively based on a measure of distance between the predicted offset and the detection in the previous image.

In an attempt to model MOT as an unsupervised learning task, Tracking by Animation [46] (TBA) was introduced. Since MOT works following the supervised learning approach, they require lots of annotated training data which can be costly to collect. This method

aimed to design a differentiable neural network to track targets where objects are "animated" into reconstructed frames. Thus, a reconstruction error is used to train the network. The authors also introduce Reprioritized Attentive Tracking to improve data association which utilizes attention to avoid overfitting. TBA achieves promising performances on some tracking benchmarks but overall not solid enough for real-world use cases.

Most recently, and taking advantage of a revolutionary architecture in natural language processing, Meinhardt et al. introduced TrackFormer [47], a multi-object tracker with Transformers. This tracking-by-attention method performs joint detection and tracking, and utilizes a Transformer network to encode features extracted from a CNN, then decode queries into bounding boxes with identities. Data association is performed using track queries, which are autoregressive representations of targets that encode spatial and temporal cues.

Summary and positioning of our work

To summarize, while one-shot/joint detection and tracking approaches achieve comparable or even state-of-the-art performances, their main disadvantage is the lack of modularity. These methods present single models trained and tailored towards a specific application. Therefore, re-training is required if we need to track new categories of objects. Tracking-by-detection approaches, on the other hand, allows us to swap the detection models or feature extractors to adapt to different tracking contexts/use cases more easily.

This latter two-step scheme has been modeled in various ways (siamese networks, CNNs, RNNs, Transformers, etc) yet, surprisingly, MOT methods with simpler techniques, such as V-IOU and DeepSORT, achieved comparable performances while maintaining high processing speeds. We have also seen the use of many types of information to describe objects of interests, including spatial, motion and appearance information. IOU-based trackers achieve a good balance between speed and accuracy. Our proposed method builds on V-IOU to take advantage of the forward and backward visual tracking, and extends data association with ReID features extracted from a model trained extensively on vehicle ReID datasets. We support the use of appearance features along with IOU for an overall more robust matching between detections.

2.2 Appearance features for tracking

In the context of MOT, describing how an object visually appears in an image is a very useful information that can be utilized for data association, especially when spatial cues are missing or inaccurate. Miah et al. [48] provided an extensive analysis of key appearance descriptors

and affinity measure in the context of MOT, which we will expand in this section.

Appearance descriptors can range from simple and traditional CNN-free techniques, all the way to deep features and ReID targeted cues. A very popular technique used in MOT is color histograms, which is a representation of how the distribution of colors in an image. In a nutshell, this technique counts the number of pixels in each color range/bin considered, without taking into account their spatial locations. This method is very simple and can be applied to input images from different color spaces. However, it does not represent information like texture or shape which are often very distinctive especially in MOT. It is also easily affected by changes in lighting.

Hence, another popular technique referred to as histogram of oriented gradients (HOG) [49] is also used in MOT methods. Unlike color histograms, HOG is able to encode information about the object structure and shape. To do that, this techniques starts by applying filter kernels to generate vertical and horizontal gradients, which lead to estimating angles and magnitudes as well. Then, it simply counts the number of occurrences of sampled angles per region, weighted by gradient magnitudes. However, this method is sensitive towards scale changes and rotations.

While CNNs gained popularity in various computer vision tasks, they inherently learn representations of objects in an image when trained to perform classification tasks. Hence, many CNN backbones can be utilized to extract deep feature descriptors of object regions. In Miah et al.'s analysis [48], four CNN models were used which have been trained on ImageNet [50] dataset and achieved solid performances. Each of the four models yields a different sized feature vector. In the same scope of deep features, a couple of ReID models were also used, one for pedestrians and one for vehicles. Findings in this work [48] suggest that, in most cases, the strongest performers for MOT are ReID feature descriptors when used to compute object similarity using the cosine metric. This conclusion is valid independently from the accuracy of the detections which makes it a great choice for data association in tracking-by-detection schemes. ReID cues are generated by models that learn whether a pair of objects are the same or not through measuring the distance between their features. A ReID network describes a vehicle appearance through images taken from different angles and/or times, hence allows us to match vehicles even as they change direction and/or spatially shift after occlusion.

2.3 Object detection

In the context of tracking-by-detection, and MOT overall, the performance of the object detection model used, directly and considerably impacts the overall tracking training process

and results. Using a robust detection method is crucial to construct consistent and accurate tracks, as well as to correctly describe the object. All current state-of-the-art object detectors are CNN-based. Therefore, we introduce some basic concepts about these neural networks then follow up with a review of object detection methods.

Convolutional neural networks are a sub-class of deep neural networks primarily applied to visual data (images and videos). Like classic feed-forward neural networks, an input is multiplied by a weight matrix, in this contexted referred to as filter kernel. The key operation here is the use of filter convolutions to capture spatial and channel-wise information in the image. CNNs contain various types of layers such as convolutional, activation, pooling and fully-connected layers. These constitute the basic building blocks of most CNN structures nowadays.

With extensive and continuous research to improve CNNs for various applications and contexts, proposed architectures evolved from early AlexNet [51] all the way to more complex structures like Inception [52, 53], ResNet [16] and stacked Hourglass networks [54].

With this in mind, a widely adopted approach to object detection divides the task in two phases. The first stage extracts Regions of Interest (ROIs) that represent target candidates, then the second stage classifies them and regresses the object location, class label and a confidence score. Following this design, the first CNN based detector named R-CNN [55] was introduced. It uses selective search, an external method, to locate RoIs then classifies these candidates by extracting their features from a deep convolutional network then applying an SVM classifier. This method is recognized as a milestone in object detection as it greatly improved performance on the task.

Later came improvements build on the R-CNN model like SPP-net [56] and Fast R-CNN [57] that enhanced inference time by extracting RoIs from feature maps, and highlighted the heavy computational load of separate region proposal methods. Hence, Ren et al. proposed Faster R-CNN [9] which reused full-image features from a region proposal network in the detection model, this way the whole image is fed to the network then portions of the feature map are fed to the classifier. The network design is presented in figure 2.6. By merging region proposal and Fast R-CNN detector into a single network end-to-end trainable, they were able to improve both performance and speed at the same time.

The introduced region proposal network in the previous work allowed to generate RoIs by regressing anchor boxes. The latter have been later adopted in many object detection models such as Mask R-CNN [10] which generate object detections and segmentation masks at the same time. It extended Faster R-CNN by adding a segmentation head similar to the bounding box regression branch, as illustrated in figure 2.7.

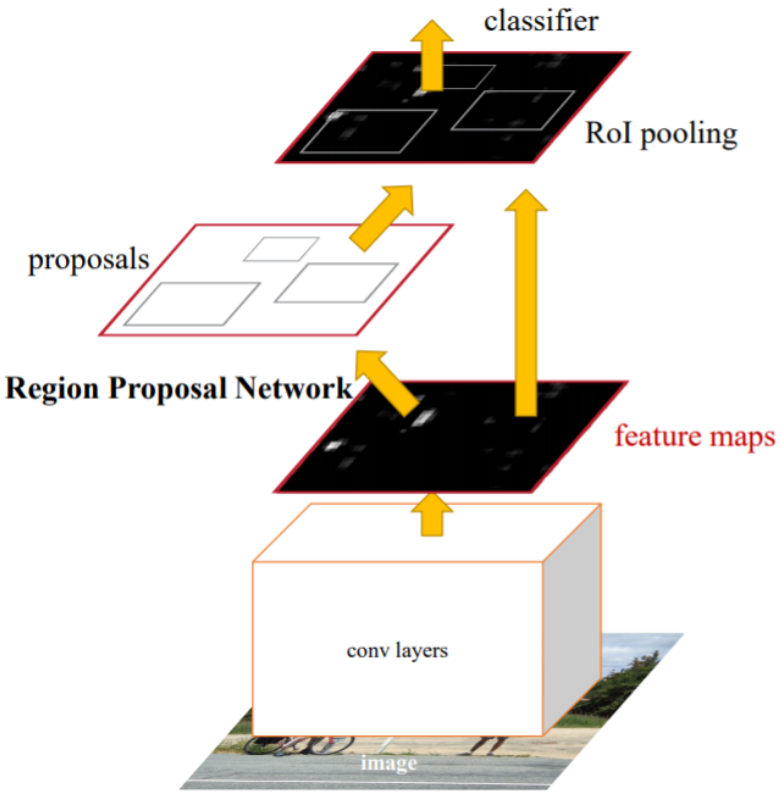


Figure 2.6 Network components of Faster R-CNN: The RPN module plays the role of an attention mechanism [9] (© 2015, IEEE).

A Region-based Fully Convolutional Network [58] (R-FCN) was proposed as a further improvement from Faster R-CNN. It replaces the fully connected layers in the box regression head by position-sensitive score maps for better detection results. Thus, parts of objects are detected then a grid is used to vote on objects. This method uses a ResNet backbone as its fully-convolutional classifier.

Another interesting approach was proposed by Lin et al. [11] which combined advantages of deep CNNs and feature pyramids from recognition systems. This work framed the problem as building multi-scale semantic feature maps to improve predictions for object detection, as illustrated in figure 2.8. The inherent hierarchy of layers in a CNN is exploited to form a top-down network flow with lateral connections to extract low-level and high level features. The features are then merged to show that the proposed Feature Pyramid Network (FPN) can be considered as a strong method for features extraction as well as a backbone for object detection models, as results show by using FPN in a Faster R-CNN system.

A different paradigm of object detection considers the task best fit by one-shot algorithms. One-stage object detectors discard the region proposal step which normally generates can-

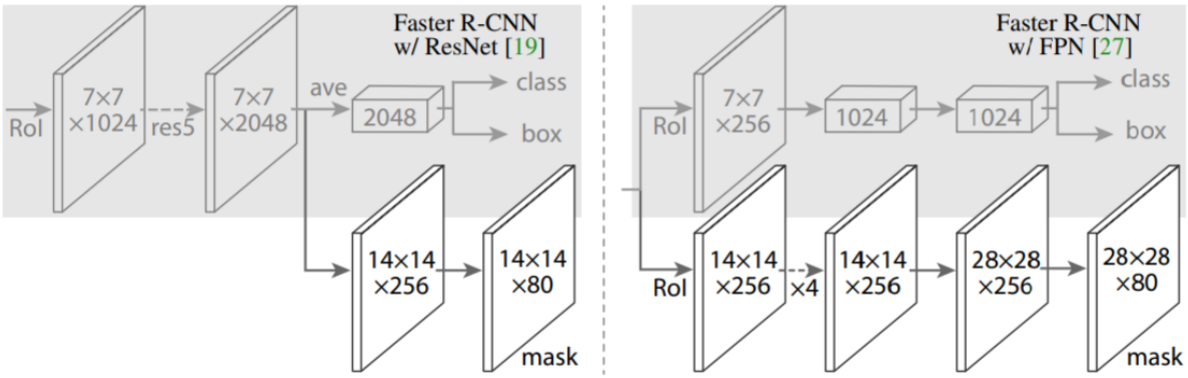


Figure 2.7 Architecture of Mask R-CNN: a segmentation branch to predict objects masks is added to the Faster R-CNN network in two configurations with different backbone CNNs (ResNet and FPN) [10] (© 2017, IEEE).

didates and instead, directly regress and classify objects from features maps which reduces processing time. The first one-step detector is known as YOLO [12] which stands for You Only Look Once. This method uses a single neural network which processes a full image only once by dividing it into a grid, then each grid cell generates two bounding box predictions. An example is used to demonstrate the grid concept in figure 2.9. Overall, fewer candidates are generated compared to previous approaches. YOLO excels greatly in speed and is end-to-end optimized, however, it struggles to detect small objects. Therefore, this unified model was later revisited in [59–61] by tweaking backbones among other things, with the latest version using an attention mechanism to train the network on focusing on important regions in the input image.

Liu et al. proposed a Single Shot multibox Detector (SSD) [37] that defines a set of default boxes over multiple feature levels to classify and regress anchor boxes. It handles detecting objects of different sizes by combining predictions from different feature maps.

The RetinaNet model [62] addresses the problem of heavy class imbalance between foreground and background by modifying classic cross-entropy loss by increasing weight of miss-classified examples. This introduced weighted loss is referred to as Focal Loss.

Some methods treat the task from a keypoint perspective, suggesting to predict important points and regress bounding boxes from them. In this scope, we mention CornerNet [63], which uses an Hourglass [54] backbone and predicts locations of the top-left and bottom-right corners of an object bounding box. Therefore, this approach does not require prior definition of anchor boxes. They also introduced a new layer called corner pooling to improve corner localization.

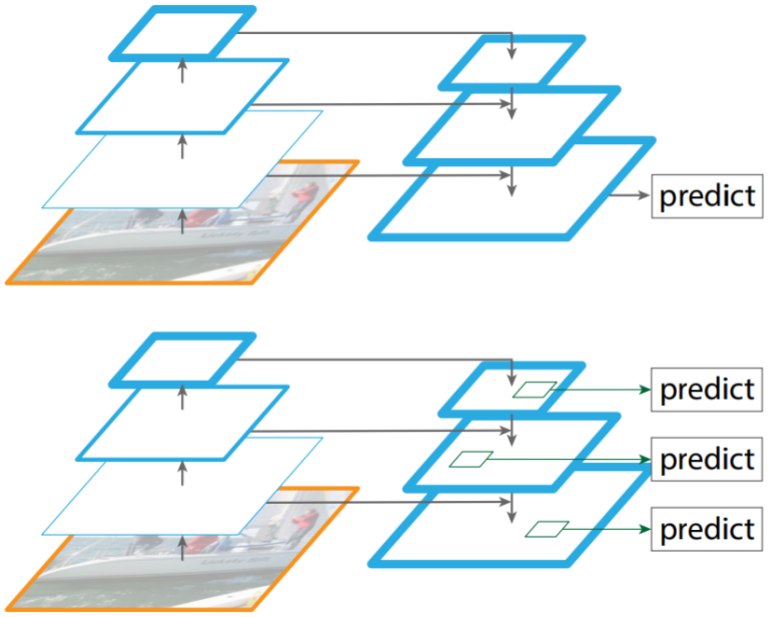


Figure 2.8 Architecture of FPN (bottom): the proposed method makes predictions at multiple levels independently, while a similar model (top) only bases the prediction on the finest level [11] (© 2017, IEEE).

While keypoint-based approaches often yield a large number of false bounding box predictions, Zhou et al. proposed a model that remedies this problem by regressing a single keypoint which is the center of the object/bounding box. This method known as CenterNet [26], or Objects as Points, predicts all of the detection attributes like width and height, 3D location, orientation and pose from its center point and matching of vertices. The output of this model is a center-point heatmap per class, object size and point offset.

Based on CenterNet, SpotNet [13] was proposed recently which uses a stacked two Hourglass networks backbone to localize object center points as demonstrated in figure 2.10. The addition is a semi-supervised segmentation branch that uses classic background subtraction/optical flow to construct segmentation masks. These segmentations are used to guide the model attention towards regions in the image that are more likely to contain objects of interest. Multi-task learning is adopted to train the model for detection and segmentation. This method achieves state-of-the-art performances on two benchmarks while keeping inference speed reasonably high for real-world systems.

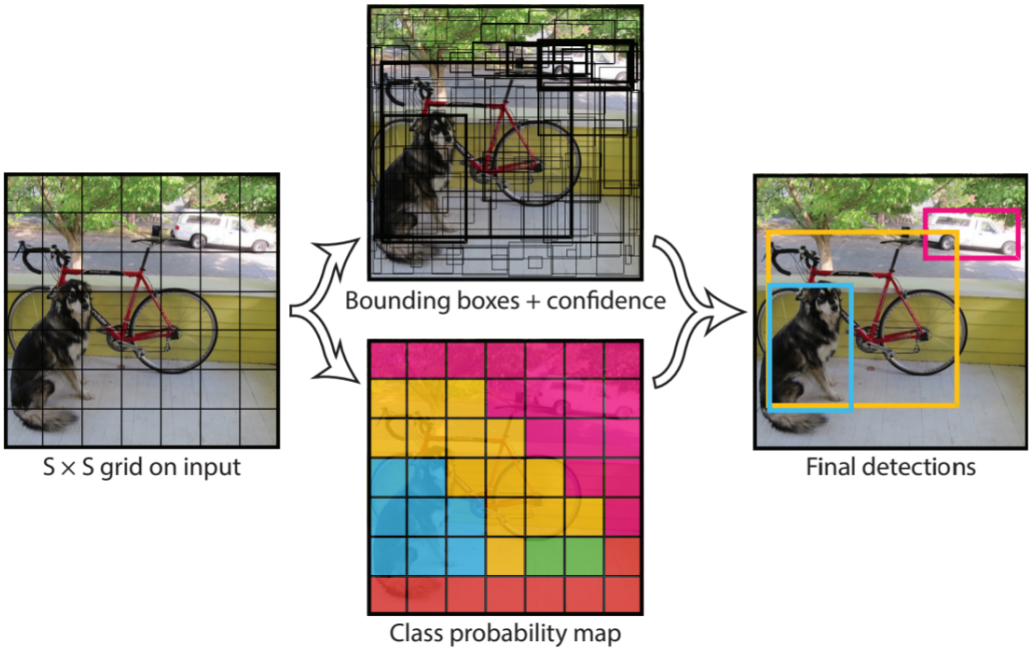


Figure 2.9 Concept of YOLO: input image is broken down into a grid and each grid cell is responsible for predicting B number of bounding boxes, confidence scores and C class labels [12] (© 2016, IEEE).

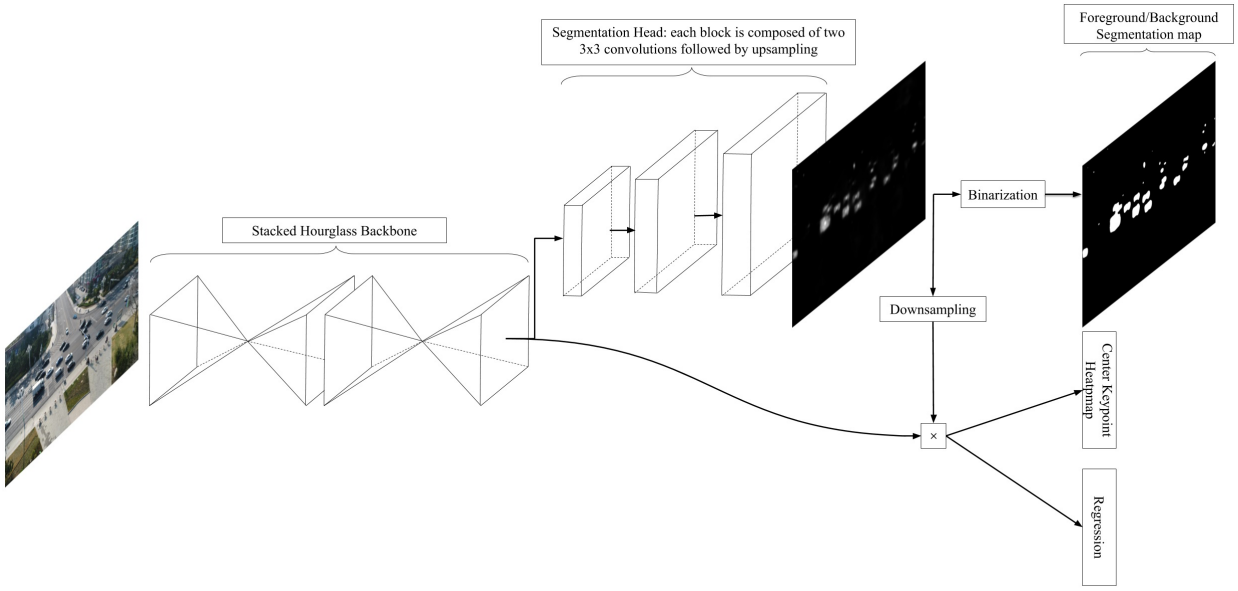


Figure 2.10 SpotNet design: the resulting feature map from the Hourglass backbone is fed to the segmentation head to generate an attention map, which is later applied to same the feature map before regressing bounding boxes as well as center point heat maps [11] (© 2020, IEEE).

CHAPTER 3 DESIGN OF THE PROPOSED SOLUTION

In this chapter of our thesis, we present the methods that we chose for both the object detection and multiple object tracking components of our system. In section 3.1, we dive into detail of our selected models and highlight their strengths and weaknesses, then in section 3.2, we introduce our proposed modification to the tracking method to improve its performance. Finally, in section 3.3, we give a complete view of our detection and tracking pipeline and its workflow.

3.1 Selected methods

The goal of our work is to localize vehicles in video sequences from surveillance cameras, then associate unique IDs to each target and maintain them throughout the video.

To tackle this problem, we proceeded with a tracking-by-detection approach given the strong performance of such methods in terms of MOT metrics and speed on various challenges and MOT benchmarks, as well as their adaptability to various applications.

This implies that our work uses two main building blocks: an object detection model to predict vehicle locations followed by a multi-object tracker to associate detections between frames.

To detect the vehicles in video frames, we chose SpotNet [13] as our detection model for a healthy balance between speed and precision. The accurate bounding boxes constitute input of the MOT method we chose: V-IOU [2] which matches detections using the IOU metric and performs visual object tracking when a track is not matched with a new detection. Below is a detailed description of each model.

3.1.1 SpotNet detector

Recently proposed by Perreault et al., SpotNet [13] is a simple and efficient object detection model using a CNN backbone and regression as well as segmentation heads. This method builds on a previous work known as CenterNet [26] which is a keypoint-based approach. A keypoint-based approach assumes that objects of interest are better detected by important and representative points, instead of modeling the detection task as a bounding box classification task. This gives a great advantage compared to classic approaches because it greatly saves on computational load by yielding fewer candidate bounding boxes, and it does not require manually designing anchor boxes for training.

SpotNet is a CNN-based architecture that employs a dual Hourglass [54] network composed of sequences of down-sampling layers and convolutional layers for the left half of the hourglass shape, which creates a dense encoding of the input, then the right half of the network applies convolutions and up-sampling operations with skip connections to construct a feature map. This category of networks is often known as encoder-decoder.

CenterNet trains this backbone CNN to recognize the keypoint of an object: its center. To do this, the center pixels of ground truth bounding boxes from the training data annotations are used as ground truth centers. Regression heads are then added which process the resulting feature maps and output the width, height and coordinate offset of the vehicle bounding box.

The novel idea in this method is the use of a self-attention mechanism to allow the model to focus on important areas in an input image. This idea is widely adopted in natural language processing and has seen great success in shifting the focus of language models towards relevant text segments. The same principle is applied in SpotNet to mimic the human way of directing attention to portions in an image that are more likely to house the object. Attention in this context is modeled by multi-task learning where object segmentation is learned as an additional task to guide the model.

Therefore, an object segmentation branch is added to the network to predict 2 classes: foreground and background. This branch is trained with semi-supervised segmentations extracted using a background subtraction model named PAWCS [64] for fixed camera configurations, while Farneback optical flow [65] was used for moving camera videos. Produced segmentations are then fitted to the ground truth bounding boxes to eliminate pixels that do not represent the objects of interest. By doing so, the segmentation head can be trained to produce foreground/background segmentations which are, then, multiplied with the features map produced by the backbone to give what is called an attention map. When visualized on an input image, the attention map resembles an image with spotlights on the objects, as shown in figure 3.1.

Finally, the attentions maps are then used to accentuate the response in feature map regions that contain vehicles, whereas the background will be "dimmed". The final training loss for SpotNet is a combination of binary cross-entropy for the segmentation head, focal loss for the heatmap head and L_1 losses for the width/height and offset heads.

SpotNet achieved state-of-the-art performance on two object detection benchmarks with traffic scenes, UA-DETRAC [1] and UAVDT [66]. We reproduced SpotNet results and also evaluated other commonly used object detection models on the UA-DETRAC benchmark to compare performances. Results will be presented and interpreted in the next chapter.

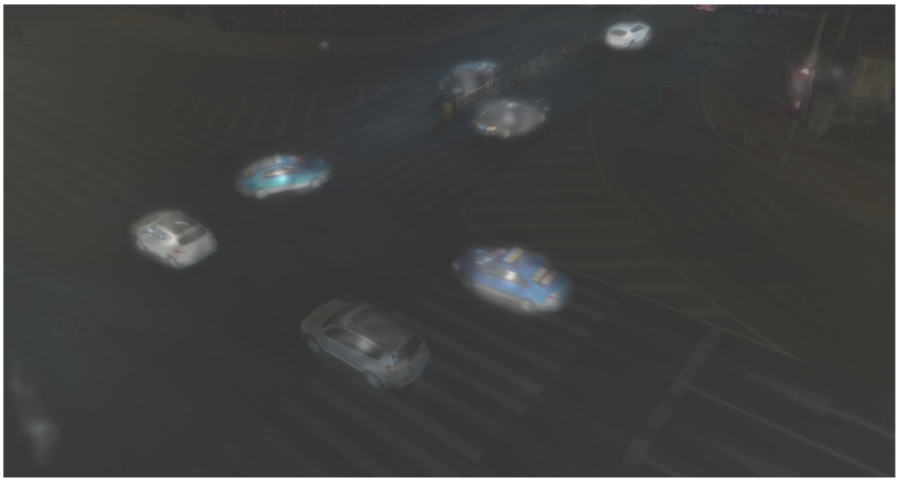


Figure 3.1 Sample attention map generated by SpotNet [13] (© 2020, IEEE).

In summary, we chose to use SpotNet as our object detector for the following reasons:

- State-of-the-art performance.
- Real-time inference speed.
- Good documentation.
- Open-source and model weights available.

3.1.2 V-IOU tracker

The V-IOU [2] MOT method proposed by Bochinksi et al. follows the tracking-by-detection paradigm, where resulting bounding boxes from the object detection model are the input of the tracking method, which backs our choice for a robust detector.

As its predecessor method [5], to link detections between frames, this method relies on the IOU metric which is demonstrated in figure 3.2. It measures the amount of overlap between the ground truth annotations and the predicted bounding boxes.

The assumption in [5] is that the input candidate detections are reliable enough. With this in mind, a greedy solver is used for the assignment, meaning that simply the highest score of spatial overlap between a track and a new detection is selected. However, the results showed that, in reality, even a few missing detections can interrupt tracks, create fragmented trajectories and lead to frequent ID switch.

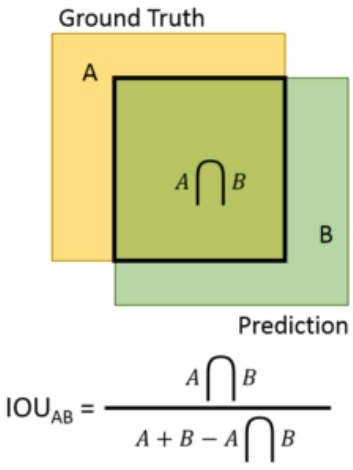


Figure 3.2 The IOU metric explained. [14]

So, to address this issue, V-IOU [2] proposed two modifications: First, it solves a linear assignment problem along with a configurable minimum IOU threshold σ_{IOU} to associate detections to tracks. Another type of filtering is also, later, applied to tracks to eliminate those with fewer than t_{min} boxes, thus requiring a minimum track length. Second, it falls back to visual object tracking when no association is found. When a track is not associated to a new detection at frame t, a KCF [67] visual object tracker is instantiated per track, and it aims to predict the object location in two directions, forward and backward.

Forward: it predicts the locations of the vehicle in the next frames until a new detection with a minimum σ_{IOU} is matched with the track thus the tracks is continued, or until the TTL counter is exhausted in which case the track is terminated. Backward: using a visual tracker forward for many frames compounds the errors in the predicted locations which means it can deviate the track away. To remedy this problem, visual tracking is also performed from frame t until a maximum of $t - TTL$ past frames in order to potentially merge with a previously terminated track. This means V-IOU is able to fill gaps in trajectories of up to $2 * TTL$.

This method achieved state-of-the-art performance of the UA-DETRAC MOT benchmark while also performing in real-time speed. The results obtained by the authors demonstrate that V-IOU reduces the fragmentation and ID switch rates.

To sum up, we chose to use V-IOU as our base MOT method for the following reasons:

- State-of-the-art performance on the UA-DETRAC vehicle tracking benchmark.
- Real-time processing speed.

- Modularity to adapt to future use cases.
- Good documentation.
- Open-source.

However, the two shortcomings of this model are the following: it scores a moderately high number of false positive likely generated by the visual object tracker predictions, and the second is that it solely uses spatial information from bounding boxes to do the association, which implicates a vulnerability in urban settings where vehicles appearances can change after occlusion. In these situations, the IOU score/bounding boxes alone may not best represent the target, and features like appearance cues add more information about the object. To this end, our proposed extension to this method uses ReID features. More detail of this contribution is in the next section.

3.2 Proposed extension with re-identification features

In order to create a more efficient and robust MOT system, we propose to extend the matching process of the chosen V-IOU tracker with information describing the vehicle visual appearance to deal with long-term occlusion and fast motion. Appearance cues have been proven to improve the association results by reducing fragmentation and ID switches.

Based on the extensive review of various appearance features for MOT in urban scenes carried out by Miah et al. in [48], we decided to extend the data association component of our MOT method with vehicle ReID features from [17]. Particularly, we solely make use of their feature extraction module trained on three vehicle ReID datasets.

In this work, Wu et al. proposed a CNN specifically trained to extract appearance features from input vehicle images of three datasets targeted for vehicle ReID. The datasets used for training are VeRi [15, 68], CompCars Surveillance [69] and BoxCars [70]. A sample collage of images from the VeRi dataset is shown in figure 3.3.

The CNN feature extractor uses a ResNet [16] architecture with 50 layers and is trained using Triplet loss and cross-entropy loss. Figure 3.4 shows the architecture of the feature extractor. Cross-entropy was used when training with classification annotations, such as with the CompCars Surveillance and BoxCars datasets, which contained labels for car model classification. Thus, the CNN learns to minimize the classification loss between car models, and this leads to learning robust discriminative features regardless of the task learned, so this can be generalized to tasks like color classification. On the other hand, Triplet loss from [71] is used with the VeRi dataset. Originally proposed for learning the similarity

Figure 3.3 Sample training images from the VeRi dataset [15] (© 2016, IEEE).

between faces, this loss function minimizes the distance between an anchor (ground truth annotation) and a positive sample (sample of the same class/label), while maximizing the distance with a negative sample (from a different class). Hence, the name triplet refers to the anchor, positive and negative. This is computed by the following equation:

$$L_{triplet} = \sum_{i}^{N} [\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha] \tag{3.1}$$

Where $f(x)$ is the feature vector of sample x, subscripts a, p, and n represent respectively the anchor, positive and negative samples, and finally α represents an acceptable bias or threshold.

The feature extraction model is also fine-tuned using an adaptive feature learning technique proposed in the same paper. This approaches aims to extract positive and training samples from unlabeled video sequences as shown in figure 3.5. This allows the network to adapt to the visual domain to test sequences. The use of this unsupervised labeling process is backed by the idea that simply a vehicle cannot be in two places at once in a video, which is referred to as space-time prior. Therefore, all surrounding vehicles in the same frame are negative samples.

By training a CNN on such crops around vehicles from different viewing angles, a robust feature extractor is created that is able to correctly describe objects regardless of their pose, illumination level and viewing angle.

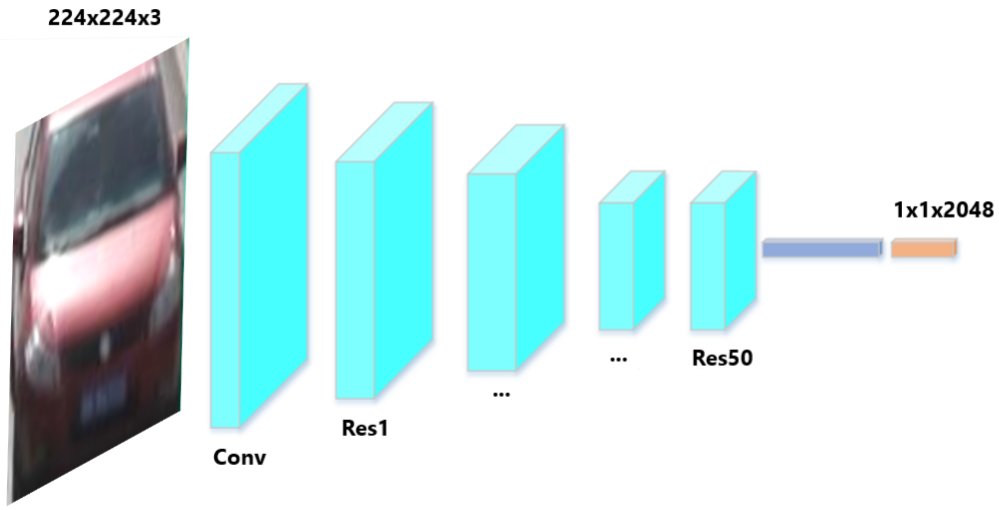


Figure 3.4 Architecture of the feature extractor used to generate 2048-dimensional vehicle descriptors based on a ResNet50 [16].

3.3 Proposed real-time detection and tracking setup

Having justified our choice for each component in the previous sections, we then propose a vehicle detection and tracking pipeline that handles scenarios with fast motion and long-term occlusions, and able to process video frames end-to-end in real-time. The following are the key steps of our system:

- Each frame is provided as input to the SpotNet object detector which returns the vehicle locations (bounding box coordinates), category and confidence score.
- For each candidate detection, the bounding box region in the image is cropped and fed to the ReID feature extractor to generate the vehicle appearance descriptor.
- Data association is performed to match detections between frames based on the IOU metric and the similarity between the ReID feature vectors. The ReID part of our cost function helps to better handle displaced bounding boxes due to high speeds, and also strengthens matching between prediction from the visual tracker and detections in occlusion scenarios.

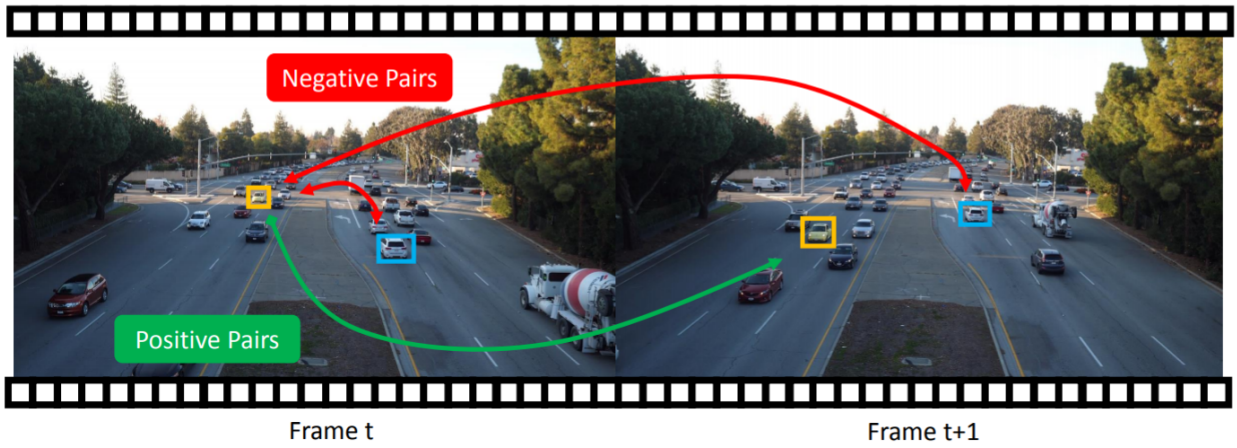


Figure 3.5 Demonstration of sample positive/negative pairs of vehicles extracted for training the feature extraction CNN [17] (© 2018, IEEE).

Our method has some similarity with previous ones, but it differs in some key aspects. The key differences between our approach and DeepSORT are:

- The use of ReID features from a CNN trained extensively on vehicle datasets, versus the deep features used in the latter method. As analyzed in [48], ReID features have shown better results on multiple MOT benchmarks compared to classic deep features which so not necessary take into account changes in viewing angles or orientation of the object.

While the approach in [3] is quite similar to ours, the following are the main differences:

- We build our method on top of V-IOU [2] which applies forward and backward predictions to reduce ID switches and fragmentation.
- We incorporate appearance features in two levels of association: first, when computing the overall cost matrix between each pair of detection/track. Second, when matching a predicted object location with unmatched tracks. This ensures that both appearance and spatial features are taken into consideration for each scenario to handle big displacement and longer occlusions.

We, now, describe in detail our proposed method. First, we collect videos from recorded surveillance cameras in urban settings such as roads, intersections and parking areas. We assume the camera to be fixed and installed high enough to have roughly an angle between 45 and 60 degrees from ground level. This gives a direct view of the moving cars without

being too high up that it becomes closer to a bird's eye view which is not a type of view used during training. The vehicle speed is also assumed to be moderate, with no extremely fast movements which can cause too much blur of location between two consecutive frames making the data association harder. It is also important that the videos are recorded at a reasonable frame rate so that the vehicles appearances do not change notably between frames.

Once vehicles bounding boxes are generated, they are passed to both the feature extractor model to generate the feature vectors representing the object appearances, and to the MOT method to carry out the data association phase. The base V-IOU tracker associates detections to tracks based solely on the IOU metric. It simply computes the overlap between each pair of candidate detection and last registered bounding box in an established track. A cost matrix is hence built with all association costs (the cost is basically $1 - IOU$). Once all IOU-based costs are computed, a fixed threshold is applied to filter out pairs which are "expensive" to match, in other words with an IOU score less than the expected threshold. Finally, a linear assignment problem solver is used based on shortest path augmentation to return the final associations to keep. This approach insures the most accurate matches are selected in a slightly faster processing time than the Hungarian algorithm. For tracks that have not been updated with a new detection, visual object tracking is performed forward and backward to predict the object location over the next and past TTL frames. This reduces the number of ID switches that occur due to short-term occlusions.

Our proposed extension comes in at this level to modify the matching scores/costs: we add to the cost a weighted cosine measure between the feature vectors of a candidate detection and a track last associated bounding box. The cosine similarity computes the cosine of the angle between the two feature vectors to tell if they have similar orientation/direction. It is calculated as:

$$cosine(x, y) = \frac{x.y}{\|x\|.\|y\|} \tag{3.2}$$

Where x, y are the vectors. The higher the value of $cosine(x, y)$, the more similar the vectors are, therefore the stronger the resemblance between the objects they describe. Thus, the cost of association between a detection d_i and a track t_j, with R_{d_i}, R_{t_j} respectively their ReID feature vectors, is formulated as follows:

$$cost(d_i, t_j) = 1 - \alpha * IOU(d_i, t_j) - \beta * cosine(R_{d_i}, R_{t_j}) \tag{3.3}$$

with $\alpha + \beta = 1$ and both are hyper-parameters to determine during find tuning to find the best values.

While visual object tracking helps solve the problem for short-term occlusions, features from ReID naturally enable handling of longer periods of occlusions because they are extracted from a CNN trained on multiple images of the same object with different orientations and lighting levels, which makes it invariant to those challenges. Thus, when an object is occluded for a long period of time, the new association function will compute the spatial and appearance similarity between the predicted bounding boxes and candidate detections. So, the object is likely correctly described by the ReID model and therefore, more likely to be retrieved after longer occlusions compared to only using IOU for matching. In scenarios where the vehicles move at relatively faster speeds, the predicted locations and the candidate detections would be distant which yields an IOU score close to zero. The ReID extension plays and important role in improving the likelihood of correctly matching predictions to detections in fast motion scenarios because it helps reduce the association costs. The ReID vectors are also used to match backward predictions to ended tracks to create more unified trajectories with less gaps. Lastly, unmatched detections initialize new tracks, and unmatched tracks are visually tracked until linked to new detections in the next TTL frames, else they are terminated.

To keep the total processing time of our system as minimal as possible, we add a custom cosine similarity function implemented in Python and accelerated using a library called Numba [72] which we will present more in detail in the next chapter. Using this Python package allows us to reduce the cosine affinity execution time to approximately $1/36^{th}$ compared to a standard implementation. This way, our ReID extension barely increases the total inference time of our pipeline, and we maintain an end-to-end processing in real-time.

Our full pipeline is summarized in the figure 3.6 and in the pseudo-algorithm 1.

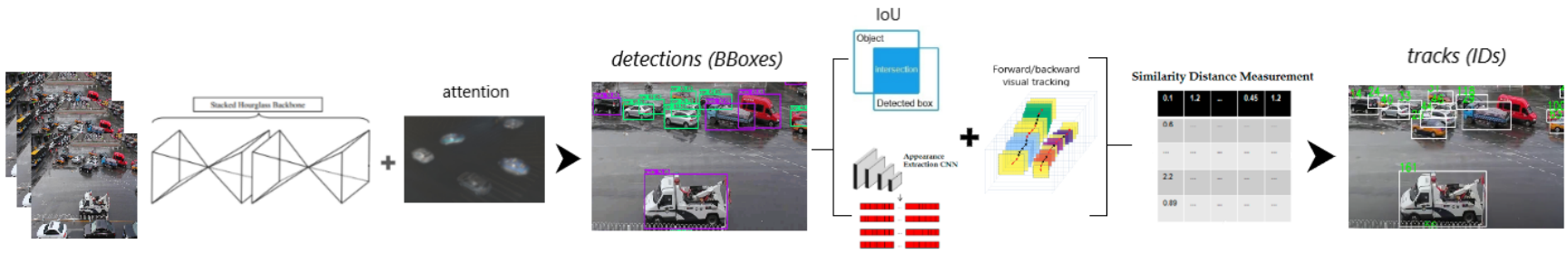


Figure 3.6 Illustration of our proposed tracking-by-detection pipeline: we modify the association step to include appearance and spatial information.

Algorithm 1: MOT algorithm with visual object tracking and re-identification

Input: detections $D = \{D_0, D_1, \ldots, D_t\} = \{\{d_0, d_1, \ldots, d_n\}, \{d_0, d_1, \ldots, d_m\}, \ldots\}$
Initialize: T_a: active tracks, T_e: extendable tracks, T_f: finished tracks
$\sigma_{IOU}, TTL, t_{min}$

for $t = 0$ *to* t **do**
  for d_i *in* D , t_j *in* T_a **do**
    $M_{cost}[d_i, t_j] = 1 - \alpha * IOU(d_i, t_j) - \beta * cosine(R_{d_i}, R_{t_j})$
  end
  $M_{cost} = M_{cost} > 1 - \sigma_{IOU}$
  track-ids, detection-ids = $lapsolver(M_{cost})$
  for d_i, t_j *in track-ids, detection-ids* **do**
    add d_i to t_j
  end
  for t_j *in* T_a*−track-ids (unmatched tracks)* **do**
    if $TTL > 0$ **then**
      add t_j to T_e
      bbox = visual-tracker(T_j) (forward)
      add bbox to T_j
      $TTL = TTL - 1$
    else
      if $length(T_j) >= t_min$ **then**
        add t_j to T_f
      end
    end
  end
  new-dets = $\emptyset$
  for d_i *in* D*−detection-ids (unmatched detections)* **do**
    bbox = visual-tracker(d_i) (backward)
    for t_j *in* T_e **do**
      if $\alpha * IOU(bbox, t_j) - \beta * cosine(R_{bbox}, R_{t_j}) >= \sigma_{IOU}$ **then**
        add bbox and d_i to t_j
        move t_j to T_a
      end
    end
    add d_i to new-dets if unmatched
  end
  for d_i *in new-dets* **do**
    initialize new tracks
  end
end

CHAPTER 4 IMPLEMENTATION AND RESULTS

In this last chapter of our thesis, we describe all the details about the implementation of our proposed method. This chapter is organized as follows: we start with an overview of the dataset and benchmark we used for the majority of our tests and evaluations. We also briefly show some of the test data provided by our industrial partner, Cysca Technologies, for some visual results. Then, we describe the protocol we followed to evaluate the different configurations of our pipeline, as well as the metrics to measure the performance of each component. In section two, we give some information about the testing environments used throughout the project. Finally, the last section is dedicated to the results for each task as well as discussion about our findings.

4.1 Dataset and benchmark for evaluation

For all our tests, visualizations and benchmarking, we used the UA-DETRAC dataset from [1] and the benchmarking toolkit they provide with the data. For a small portion of tests, we used some video sequences recorded from Cysca private parking lot as a way to visually validate our method in a different urban setting that is not included in the UA-DETRAC train and test data.

4.1.1 UA-DETRAC

UA-DETRAC [1] is a dataset and benchmark suite for multi-object detection and tracking in real-world difficult scenarios. The video sequences were collected from multiple urban settings in China, at a frame rate of 25 FPS and a 960 x 540 pixels resolution. In total, it contains 10 hours of recordings, yielding more than 140,000 frames. The vehicles in these images were annotated manually every 5 frames, with linear interpolation in between, to obtain 1.21 million bounding boxes from 8250 vehicles. The annotations present the following attributes:

- Bounding box coordinates.
- vehicle type: Car, Bus, Van, Other.
- Weather condition: Sunny, Cloudy, Rain, Night.
- Scale in square root of bounding box area: small (0-50 pixels), medium (50-150 pixels), large (more than 150 pixels).

- Occlusion level: partial occlusion (1%-50%), heavy occlusion (more than 50%).

For object detection, the sequences are also categorized by difficulty level (easy, medium, hard). Hence, the dataset covers a wide range of challenging scenarios for detection and tracking, with different weather conditions, implying lighting changes, occlusions, viewing angle and scale changes. The samples in figure 4.1 demonstrate the available annotations and attributes. Fine-Tuning was conducted on the training set. For our visualizations and evaluation, we used the test set of 40 sequences.

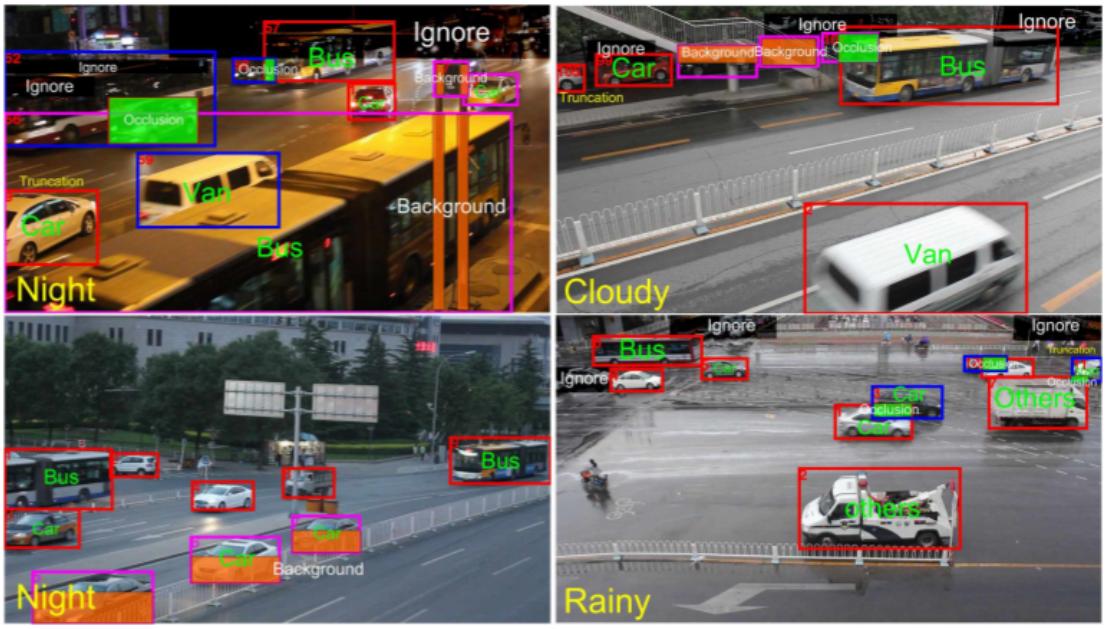


Figure 4.1 Collage of sample frames from the UA-DETRAC dataset showing the available annotations. Regions within black rectangles labeled "Ignore" are considered as background. The bounding box colors red, blue and pink represent different levels of occlusion [1] (© 2020 Elsevier Inc. All rights reserved.).

4.1.2 Cysca's test videos

As mentioned previously, we also used some sample videos recorded from the parking lot of Cysca in Repentigny. The videos are not annotated, therefore the use of these sequences was solely intended to visually validate our pipeline on new unseen urban settings. Figure 4.2 shows an example: there are two cameras installed in the parking lot referred to as the west camera and north camera, both providing complementary coverage of a large parking area. The videos were recorded at 15 FPS with a resolution of 1280 x 720 pixels.

Figure 4.2 Sample frames from the parking lot of Cysca videos.

4.1.3 Evaluation process and metrics

To evaluate our proposed method, we used the UA-DETRAC official benchmark toolkit. The tool is a set of Matlab scripts with some Windows executable files. The proposed evaluation protocol processes detection and tracking results for each threshold in the range 0.0 to 1.0 with a step of 0.1 to consider the effect of the detection model on tracking. This means that for each iteration, the detections with confidence scores lower than or equal to the current threshold are selected, then the MOT method is evaluated for this subset.

For object detection, a precision-recall (PR) is generated along with results in terms of the following categories and metrics: a detection is counted as correct if it scores an overlap with the ground truth annotation greater than 70%. The average precision (AP) per difficulty level and weather condition is then computed and used to rank the detection method.

As for MOT evaluation, the benchmark supports two sets of metrics: CLEAR-MOT metrics which consider a single detection threshold, and DETRAC-MOT metrics which cover all 11 detection thresholds between 0 and 1.

CLEAR-MOT uses the following measures: mostly tracked (MT), mostly lost (ML), identity switches (IDS), fragmentations of target trajectories (FM), false positives (FP), false negatives (FN), MOT accuracy (MOTA), MOT precision (MOTP). The FP metric measures how many false alarms a tracker generates measured with the 0.7 theshold, whereas the FN metric measures how many targets any monitored trajectories miss in each frame. The IDS metric measures the number of times a tracked trajectory matching identity changes, while the FM metric measures the number of times trajectories are interrupted. The accuracy of monitored trajectories is reflected in both IDS and FM measures. Based on the ground truth, the ML and MT metrics describe the percentage of trajectories that are less than 20% and more than 80% of the time tracked, respectively.

For all sequences in the benchmark, the MOTA measure in % is defined as follows:

$$MOTA = 100 * (1 - \frac{\sum_v \sum_t (FN_{v,t} + FP_{v,t} + IDS_{v,t})}{\sum_v \sum_t GT_{v,t}}) \tag{4.1}$$

where the pair (v, t) denote time t in sequence v, and GT represents the number of ground truth targets. The FN, FP and IDS scores used are from the tracking results. The value of MOTA is typically between 0 and 100%, though it technically can be below zero.

Additionally, The MOTP metric formulated as the average overlap between all correctly matched hypotheses and corresponding ground truths. As a result, it reflects he accuracy of the detection method, and provides little information about the tracking method. It is defined in [36] by:

$$MOTP = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t} \tag{4.2}$$

where c_t stands for the number of correct matches in frame t, and $d_{t,i}$ represents the overlap between the bounding boxes of target i and its matched ground truth object. The value of MOTP is typically between 50 and 100%.

While the previously presented metrics of MT, ML, IDS, FM, FP, FN and MOTA are a good way to reflect the tracker performance, they do not allow for a fair comparison between different tracking methods because they do not significantly take into consideration the detector performance which could vary from one method to another. In other words, the UA-DETRAC proposed a modification to those metrics that reflects the performance of the MOT method under various detection thresholds. These new metric are PR-MOTA, PR-MOTP, PR-MT, PR-ML, PR-IDS, PR-FM, PR-FP and PR-FN. All of these metrics are based on the detector precision-recall curve.

As illustrated in figure 4.3, for a subset of detections determined by a pair of precision and recall values (p, r), the tracking method is applied then the MOTA is computed. By performing this over a range of 11 thresholds, we obtain the PR-MOTA curve shown in red in the figure. Finally, the PR-MOTA measure is obtained by computing the integral of the curve as follows:

$$PR\_MOTA = \Omega^* = \frac{1}{2} \int_C \Psi(p, r) ds \tag{4.3}$$

To evaluate the performance of our proposed method and rank it against the state-of-the-art, we mainly use the PR-MOT metrics with a secondary resort to MOT metrics for more

insights about our results.

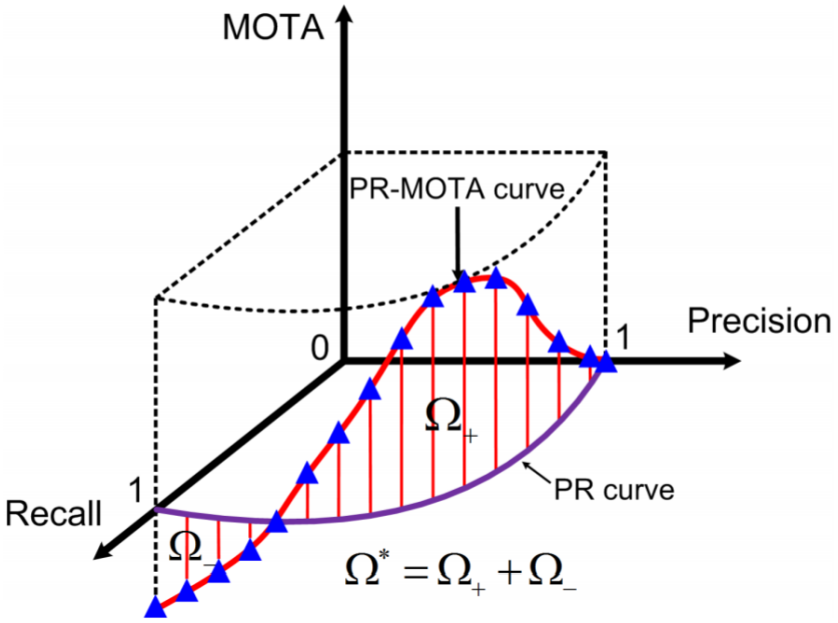


Figure 4.3 PR-MOTA curve: sampling points represented by the blue triangles are used to generate the PR-MOTA curve. [1]

4.2 Implementation and environment

To implement our proposed pipeline, we primarily used Python as our programming language of choice due to its popularity in the deep learning and computer vision community. This makes it very well documented with a large number of open source libraries and packages to improve the code. Such is the case with OpenCV [73] which was a key building block for handling the input/output as well as visualization and pre/post-processing.

Throughout the course of this project, our test environment changed. When we were evaluating and reproducing object detection results, we used a server equipped with a Nvidia GTX 1050 Ti GPU and 32GB of RAM. Later on, when implementing the modification to the tracking method, we switched to using a remote server provided for research by Compute Canada. Job submitted on their compute nodes often used V100 GPUs and 16GB of RAM. Then, towards, the end of the project, we improved the specs of the initial server with a RTX 6000 GPU. Therefore, in the detection results section below, the reported speed were from the initial server configuration and only serve as a comparison between the different detectors. The final object detector speed described later in this chapter has been reported with the latest GPU (RTX 6000).

To find the best parameter values for our proposed method with SpotNet and ReID features, we fine-tuned our MOT method on the UA-DETRAC training set with the following pools of values, mainly determined by previous configurations of the V-IOU method:

- σ_{IOU} threshold for IOU association: $\{0.4, 0.5, 0.6, 0.7\}$
- TTL visual tracking lifespan: $\{6, 15\}$
- t_{min} minimum accepted track length: $\{3, 13\}$
- α, β weights for spatial and appearance cues in data association: $\{(0.7, 0.3), (0.75, 0.25)\}$

For the choice of the visual tracking method, unlike the original V-IOU work which used a particular implementation of KCF [67], we opted for a MedianFlow tracker [74] for two reasons: the first being that the KCF implementation used required the installation of a custom python wrapper to alter original implementation of KCF in OpenCV, which was time-consuming and complex to install. Second, our tests with MedianFlow showed negligible difference compared to the KCF resuls as in table below.

Table 4.1 Impact of visual tracking method on UA-DETRAC MOT results

Detector	MOT method	Visual tracker	PR-MOTA	PR-MOTP
Mask R-CNN	V-IOU	KCF	30.3714	37.885
Mask R-CNN	V-IOU	MedianFlow	30.3584	37.9016

4.3 Results and discussion

In the last section of this chapter, we present our results and findings for both the object detection and MOT tasks. For each tasks, a subsection summarizes the results in terms of metrics, visualizations and a discussion is made to interpret them.

4.3.1 Object detection results

As we have covered previously in this thesis, choosing a strong object detector in the context of a tracking-by-detection scheme is of high priority. Therefore, we started this project with an empirical analysis of the performance of various object detectors on the UA-DETRAC benchmark for this task. Besides comparison, we performed the evaluation at this step to study the impact of some variations in the input data and/or the model weights.

As detailed in table 4.2, we compared the performance of our detector of choice SpotNet with its base model CenterNet, YOLOv4 with the attention mechanism and finally Mask R-CNN which was used with the V-IOU original method. Also, in the same table, SpotNet labels refers to the version of SpotNet that is trained with the four vehicle labels, as opposed to the binary models that detects object/background labels only. In the *Resized* column, *No* stand for passing the UA-DETRAC images in full resolution to the detector, while *Yes* means the input has been resized to 512 x 512 to save on computation and increase speed.

As expected, SpotNet is by far the top performing model on the UA-DETRAC object detection benchmark. It outperforms all tested detectors consistently across all evaluated challenges. On the GTX 1050 Ti GPU and with the full image resolutions of 960 x 540 from UA-DETRAC, we obtain the best AP score compared to using downsized images which only improved inference speed by a factor of 1.5. Another important observation is while the UA-DETRAC evaluation protocol for object detection does not take into account the object labels, the difference in performance between the model trained on binary labels (background/object) vs training on all four vehicle types is negligible, with less than 1% decrease in AP in all evaluation scenarios.

In our experiments on the latest RTX6000 GPU, SpotNet achieves an inference speed of up to 8 FPS which is very satisfactory in the application context of Cysca given the video recordings at 15 FPS.

Therefore, we proceed to fine-tune our proposed MOT method using detection results from the labeled version of SpotNet.

Table 4.2 Detection results on the UA-DETRAC test set. **Bold**: best performance. labels*: version of the model with all DETRAC vehicle categories. Resized: No for full resolution input images, Yes for resized input to 512 x 512 pixels. Speed: reported with a GTX 1050 Ti GPU.

Model	Resized	Overall	Easy	Medium	Hard	Cloudy	Night	Rainy	Sunny	Speed
SpotNet	No	**86.80%**	**97.58%**	**92.57%**	**76.58%**	**89.38%**	**89.53%**	**80.93%**	**91.42%**	1.34 fps (0.745s)
SpotNet labels*	No	85.49%	97.31%	91.78%	74.35%	88.37%	88.91%	78.27%	90.61%	1.34 fps (0.745s)
SpotNet	Yes (512)	81.63%	94.73%	88.19%	69.79%	83.23%	85.20%	75.59%	87.77%	2 fps (0.5s)
SpotNet labels*	Yes (512)	78.19%	91.37%	86.06%	64.9%	79.28%	83.86%	71.46%	82.72%	2 fps (0.5s)
CenterNet	No	83.48%	96.50%	90.15%	71.46%	85.01%	88.82%	77.78%	88.73%	-
Mask R-CNN	No	80.47%	94.32%	85.92%	69.13%	82.48%	84.26%	75.72%	81.07%	-
YOLOv4	No	72.94%	89.51%	79.82%	59.27%	73.03%	79.59%	64.81%	81.75%	0.63 fps (1.58s)

4.3.2 MOT results

To be able to rank our proposed extension to the V-IOU method as well as examine the effect of both substituting the detector by SpotNet and adding the ReID features, we start by evaluating the MOT results using the original V-IOU with SpotNet and MedianFlow. Then, we add the cosine similarity component at the data association level.

Given that the evaluation process is very time-consuming, averaging around 48 hours for a single configuration of the pipeline, we narrowed down our search space for the tracking parameters by looking at the values used in similar works and in the previous V-IOU version. After defining the ranges to test, we performed fine-tuning for the parameters with V-IOU and SpotNet on the UA-DETRAC training set to arrive at the best values to carry over when implementing the ReID extension.

Below are the best parameter values obtained after fine-tuning:

Table 4.3 Best configuration for V-IOU with SpotNet

σ_{IOU}	α	β	TTL	t_{min}
0.6	0.7	0.3	15	3

We compare the V-IOU and SpotNet combination against the state-of-the-art performance in UA-DETRAC which is V-IOU with Mask R-CNN. We also report the results of a few other top performers and baseline methods as a reference in table 4.4. We note that in this table, the results were taken from the official UA-DETRAC benchmark as well as previously reviewed papers.

Similar to the observation in [3], we noticed that SpotNet is rather "too safe", in the sense that the detections confidence scores are heavily concentrated between 0% and 80-85%. So, SpotNet rarely yields detections with confidence higher than 90%. This creates a problem when running the evaluation protocol from UA-DETRAC because the MOT method is evaluated on each threshold between 0% and 100%. This results in zeros for all MOT metrics on the last evaluation which selects detections stronger than 90%. Hence, this has a big negative impact of our tracking results. Also, the Mask R-CNN detections used with V-IOU contain the top 50 bounding boxes for each frame, unlike the top 100 provided with SpotNet. To remedy this, we resorted to a simple tweak: we selected the top 50 candidate detections in each frame, then their confidence scores were multiplied by a factor of 1.25 and clipped to 100%. This ensures a more even distribution of scores without affecting the bounding boxes themselves.

The results show that our proposed method outperforms the current top one on the UA-

DETRAC tracking benchmark, V-IOU with Mask R-CNN, particularly in terms of PR-MOTP with an increase of 13.9%. We also achieve a marginal improvement for PR-MOTA with an increase of 0.5%. Hence, we get state-of-the-art results of 31.2% PR-MOTA and 50.9% PR-MOTP. While the base V-IOU method still has the lead in terms of PR-MT, PR-IDS, PR-FM, we outperform the previous work in terms of PR-ML, PR-FP and PR-FN. Overall, our method is more consistent and better performing on most PR-MOT metrics.

Our method achieves much less false positives, nearly 3 times less than the PR-FP of V-IOU with Mask R-CNN. This reflects the significant improvement made by switching the object detector to SpotNet, which is a much more robust model and can handle various challenging scenarios. We also score a small decrease in false negative rates.

We achieve a noticeable improvement on PR-ML from 22.6 to 18.5. As described in section 4.1.3, this metric measures the ratio of ground truth tracks that are tracked 20% or less or the time. This supports the idea that combining visual object tracking with ReID features is an effective strategy for better maintaining trajectories, thus resulting in overall more consistent tracks. Nonetheless, the experiments conducted to fine-tune the weights of the spatial and appearance features for data association may not be sufficient and we leave room for more tests with different values.

To have a more detailed view about the impact of our extension on the overall performance, we compare its metric values with results obtained with SpotNet and the original V-IOU method in table 4.5. Indeed, using the ReID features significantly reduces the number of ID switches and fragmentation compared to using SpotNet with the base version of V-IOU. The greatest improvement in PR-MOTA and PR-MOTP are achieved by adding the ReID component, while the improvement in terms of PR-FP comes from using spotNet. This is what drives the improvement in terms of PR-MOTA since the tracker is now less likely to incorrectly switch IDs between two targets crossing each other's paths.

The way we interpret the increase in terms of PR-MOTP is through the design of our cost function. Our cost accounts for 30% of the ReID features similarity. In scenarios of relatively long occlusions, the longer we visually track an object with Medianflow the more error is introduced into the predicted boxes. Hence, IOU would give small scores between the predicted location of the vehicle and candidate detections. The ReID scores will fairly compensate for the IOU by reducing the association cost which makes our method able to fill gaps in trajectories.

For small objects that move at higher speeds, it appears that SpotNet generates detections that are spatially distanced. With the base V-IOU tracker, this would results in very short tracks of length one which are discarded by the evaluation protocol. When ReID features

are introduced, we are able to connect these detections despite the displacement which leads to two findings: first, this means we have less tracks deleted during benchmarking and more correct detections preserved. Consequently, we obtain less false positives and better PR-MOTP results. Second, the ReID allows our tracker to use less visual tracking results since we are able to reconnect actual detections. This also contributes to reducing the PR-FP rate.

Therefore, the advantage of our ReID extension is that it allows us to use the visual tracker a lot less, meaning less estimated locations, which yields more accurate results.

Next, we demonstrate some sample output frames from our complete pipeline. We confirm visually in figure 4.4 that out method yields consistent tracks through time. Target #31 is detected even when it first appears with more than 50% occlusion by the nearby tree. Then its ID is maintained as it changes orientation in the next 70 frames. The surrounding vehicles also keep the same IDs as their scale changes by moving further away of closer to the camera.

In figure 4.5, we show a collage of 4 sampled frames from sequence 39361 where there is a dense presence of targets. In this scene, vehicles #30 and #2 partially overlap in the span of roughly 30 frames yet their IDs remain correct. However, a very challenging scenarios happens where targets #30 almost fully obstructs the view of car #50. Thus, ID 50 is mistakenly assigned to the first car that was originally assigned #30, and vehicle #50 taken on a new ID when it reappears. This is due to a near-perfect IOU score between the two object bounding boxes.

Table 4.4 DETRAC-MOT results on the UA-DETRAC test set with top 50 SpotNet detections. **Bold**: best result in each metric. Results * taken from [1]. Results & taken from [2]. Results # taken from [3].

Detector	**Tracker**	**PR-MOTA**	**PR-MOTP**	**PR-MT**	**PR-ML**	**PR-IDS**	**PR-FM**	**PR-FP**	**PR-FN**
DPM	GOG*	5.5	28.2	4.1	27.7	1873.9	1988.5	38957.6	230126.6
EB	KIOU*	21.1	28.6	21.9	**17.6**	462.2	712.1	19046.9	**159178.3**
M-RCNN	V-IOU&	30.7	37.0	**32.0**	22.6	**162.6**	**286.2**	18046.2	179191.2
SpotNet	Kalman+IOU#	30.6	42.7	-	-	3634	-	-	-
SpotNet	V-IOU+ReID (ours)	**31.2**	**50.9**	28.1	18.5	252.6	329.1	**6036.4**	170700.6

Table 4.5 Ablation study on the UA-DETRAC test set. Bold: best result in each metric. Results & taken from [2].

Detector	**Tracker**	**PR-MOTA**	**PR-MOTP**	**PR-MT**	**PR-ML**	**PR-IDS**	**PR-FM**	**PR-FP**	**PR-FN**
M-RCNN	V-IOU&	30.7	37.0	**32.0**	22.6	**162.6**	**286.2**	18046.2	179191.2
SpotNet	V-IOU	27.6	39.8	23.6	20.9	293.1	382.4	**4783.9**	194784.4
SpotNet	V-IOU+ReID	**31.2**	**50.9**	28.1	**18.5**	252.6	329.1	6036.4	**170700.6**

Finally, in terms of processing speed, we report the following values for each component of our system on the RTX 6000 with a threshold for detections set to 0.7 (yielding up to of 20-25 candidates per frame):

- SpotNet: up to 8 FPS.
- V-IOU with ReID features/cosine: up to 60 FPS.
- End-to-end time: up to 6 FPS.

The convention for this project with Cysca Technologies defined the threshold for real-time processing to be of 1 FPS. Hence, our proposed method is more than satisfactory in terms of speed.

Computing the cosine similarity between two 2048-dimensional feature vectors can be costly. Therefore, we implemented the cosine function using Numba [72]. It is a free and open source JIT compiler that converts a portion of Python and NumPy code to machine code. This allows us to achieve a good speed up for computing the cosine between two descriptors. In our experiments with random 512-dimensional vectors, Numba took the execution time of the cosine similarity from an average of $35.5\mu s$ over 10,000 iterations, to $0.987\mu s$ over 100,000 loops. This is a major speed up as the number of targets increases in crowded scenes.

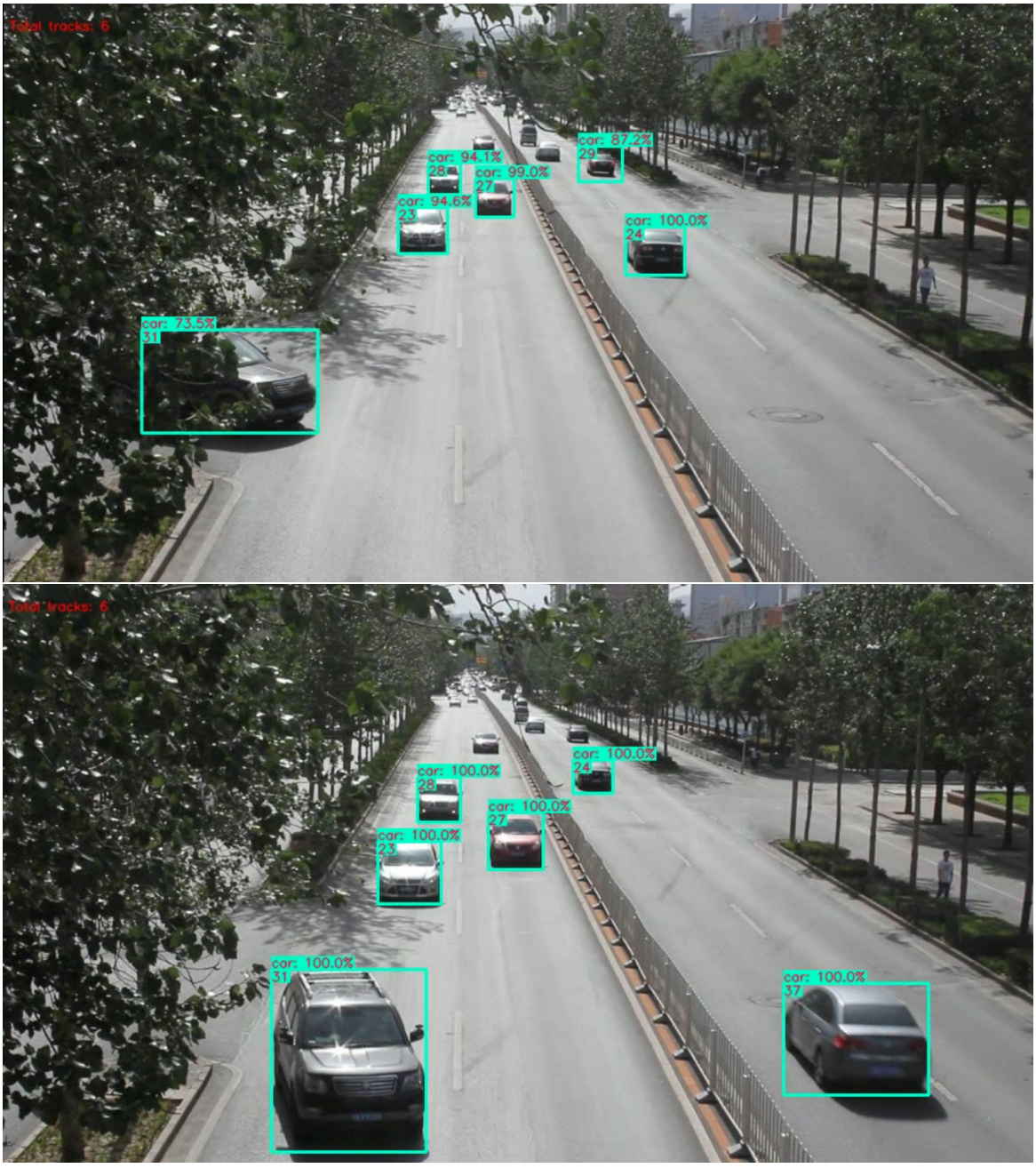
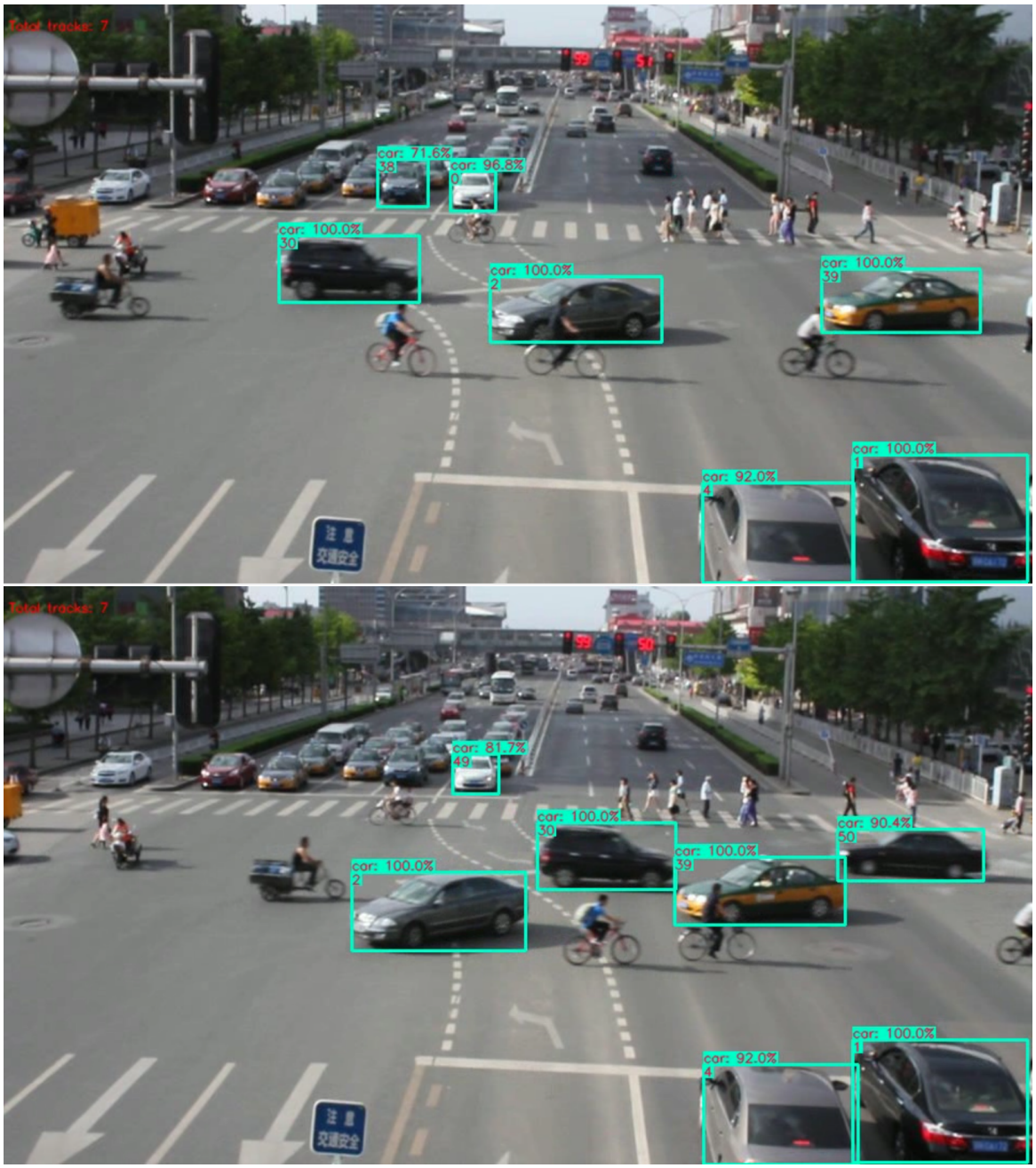


Figure 4.4 Sample output from UA-DETRAC test sequence 39031: the two scenes are 70 frames apart.


Total tracks: 7
car: 71.6%
38
car: 96.8%
0
car: 100.0%
30
car: 100.0%
2
car: 100.0%
39
car: 92.0%
4
car: 100.0%
1
注 意
交通安全
Total tracks: 7
car: 81.7%
49
car: 100.0%
30
car: 100.0%
2
car: 100.0%
39
car: 90.4%
50
car: 92.0%
4
car: 100.0%
1
注 意
交通安全

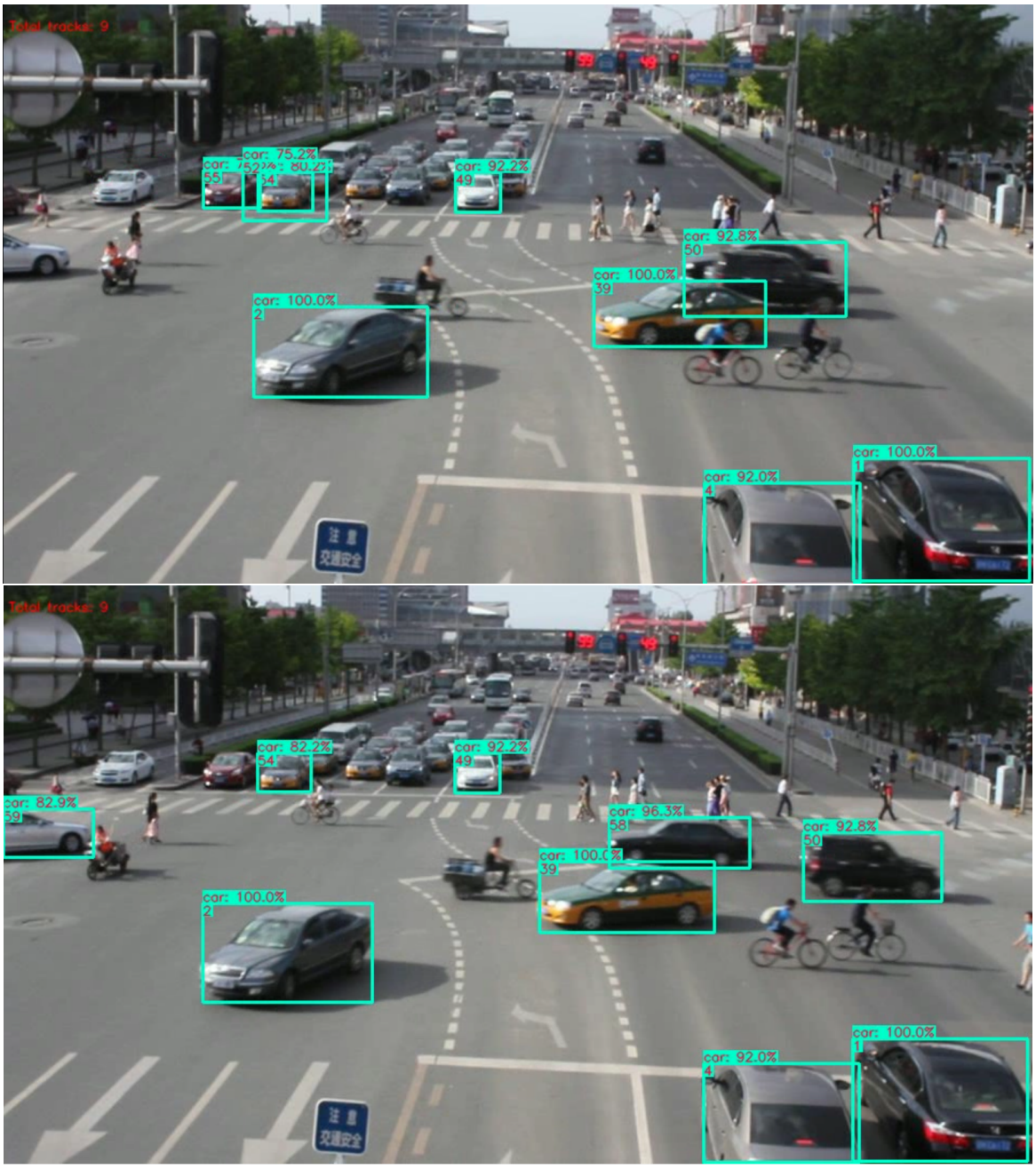


Figure 4.5 Sample output from UA-DETRAC test sequence 39361 in nearly 100 frame span: the two vehicles #30 and #50 switch IDs due to heavy occlusion.

CHAPTER 5 CONCLUSION

In this chapter, we summarize our work by restating what the contributions are, highlighting the limitations of our approach and proposing future improvements and ideas.

5.1 Summary of works

Our work presented a vehicle detection and tracking system following the tracking-by-detection paradigm. We utilized a state-of-the-art object detection model to extract the vehicle locations. Next, we build on a MOT method which uses the IOU metric for data association as well as visual object tracking to replace missing detections for a short-term. This method scores a good balance between accuracy and speed, and the key advantage is the modularity which allows it to easily adapt to new tracking applications or benefit from future improved detectors and/or feature extractors. We extend this method with ReID features and cosine affinity to reconnect detections/tracks in fast motion scenarios and better handle long-term occlusions. Our results show that indeed the performance has been noticeably improved on the UA-DETRAC benchmark. Particularly, we design the cost function to associate between detections and tracks by using a weighted sum of the IOU score between the bounding boxes and the cosine similarity between ReID feature vectors. This compensates for small IOU values when bounding boxes are shifted. The results show that the ReID component improves the overall tracking results significantly.

5.2 Limitations

Although the overall speed of our pipeline is considered real-time, it can decrease with more objects to track since the ReID features are extracted for each object. The more objects of interest there are in a scene, the more computation required to associate and track, therefore it takes longer in time to process. Luckily, the UA-DETRAC dataset contains some very cluttered real-world scenes so this gives us a realistic idea of what to expect when using our pipeline is used in a monitoring application. There is also room for tracking performance improvement particularly with longer occlusion scenarios where both the IOU and ReID similarity can fail.

5.3 Future research

As a work around for the previously mentioned problem, we can think about limiting the search space for the association with ReID. The idea is to create a "neighbourhood" of candidates that are more likely to be matched with a detection/track. This neighbourhood can have a fixed size, either a fixed search radius or a fixed number of members.

Along the same lines to improve inference speeds, we can explore some model compression techniques to preserve the performance while reducing the sizes of our models. These techniques bring the added benefit of making the models robust against adversarial attacks, and can also be explored to make the models fit for edge-computing on embedded systems.

Another perspective is to use additional information for association. These features can include the vehicle label/type and/or semantic segmentation. Using segmentation for association basically means instead of computing the IOU between bounding boxes which contains a small portion of the background, we only use pixel-level information by considering the number of pixels in the intersection of two segmentations, which can help in the scenarios displayed in figure 4.5. We can also potentially limit the extraction of the ReID features to the objects pixels as well using the segmentations, thus we would obtain ReID vectors for the foreground object only. This has the potential of improving the association both in short-term and long-term contexts.

Finally, other MOT benchmarks have already adopted a new evaluation metric called HOTA [75] which balances out the often remarkable difference MOTA and MOTP scores. Hence, it would be intuitive to incorporate this metric in our evaluation protocol in the future.

REFERENCES

[1] L. Wen, D. Du, Z. Cai, Z. Lei, M.-C. Chang, H. Qi, J. Lim, M.-H. Yang, and S. Lyu, "Ua-detrac: A new benchmark and protocol for multi-object detection and tracking," 2020.

[2] E. Bochinski, T. Senst, and T. Sikora, "Extending iou based multi-object tracking by visual information," in *IEEE International Conference on Advanced Video and Signals-based Surveillance*, Auckland, New Zealand, Nov. 2018, pp. 441–446. [Online]. Available: http://elvera.nue.tu-berlin.de/files/1547Bochinski2018.pdf

[3] Z. Kang, "Multiple object tracking in videos," Master's thesis, Dep. of computer engineering, École Polytechnique de Montréal, Montréal, QC, 2021.

[4] W. Li, J. Mu, and G. Liu, "Multiple object tracking with motion and appearance cues," 2019.

[5] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," in *International Workshop on Traffic and Street Surveillance for Safety and Security at IEEE AVSS 2017*, Lecce, Italy, Aug. 2017. [Online]. Available: http://elvera.nue.tu-berlin.de/files/1517Bochinski2017.pdf

[6] H. Kieritz, W. Hübner, and M. Arens, "Joint detection and online multi-object tracking," in *2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2018, Salt Lake City, UT, USA, June 18-22, 2018*. IEEE Computer Society, 2018, pp. 1459–1467. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018_workshops/w29/html/Kieritz_Joint_Detection_and_CVPR_2018_paper.html

[7] L. Leal-Taixé, C. C. Ferrer, and K. Schindler, "Learning by tracking: Siamese cnn for robust target association," 2016.

[8] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 107–122.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016.

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," 2018.

[11] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," 2017.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016.

[13] H. Perreault, G.-A. Bilodeau, N. Saunier, and M. Héritier, "Spotnet: Self-attention multi-task network for object detection," 2020.

[14] J. Barker and A. Gray. (2016) Exploring the spacenet dataset using digits. [Online]. Available: https://developer.nvidia.com/blog/exploring-spacenet-dataset-using-digits/

[15] X. Liu, W. Liu, H. Ma, and H. Fu, "Large-scale vehicle re-identification in urban surveillance videos," in *2016 IEEE International Conference on Multimedia and Expo (ICME)*, 2016, pp. 1–6.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[17] C.-W. Wu, C.-T. Liu, C.-E. Chiang, W.-C. Tu, and S.-Y. Chien, "Vehicle re-identification with the space-time prior," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 121–1217.

[18] Rostyslav Demush. (2019) A brief history of computer vision (and convolutional neural networks). [Online]. Available: https://hackernoon.com/a-brief-history-of-computer-vision-and-convolutional-neural-networks-8fe8aacc79f3

[19] Nick G. (2021) How many iot devices are there in 2021? [Online]. Available: https://techjury.net/blog/how-many-iot-devices-are-there/

[20] Cisco. (2011) Global internet traffic projected to quadruple by 2015. [Online]. Available: https://newsroom.cisco.com/press-release-content?type=webcontent&articleId=324003

[21] J. M. Corchado, "Efficiency and reliability in bringing ai into transport and smart cities solutions," in *International Conference on Transport and Smart Cities*, December 2019.

[22] LDV Capital. (2017) Ldv capital insights 2017. [Online]. Available: https://www.ldv.co/insights/2017

[23] Y. Qian, L. Yu, W. Liu, and A. G. Hauptmann, "Electricity: An efficient multi-camera vehicle tracking system for intelligent city," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[24] R. Szeliski, *Segmentation*. London: Springer London, 2011, pp. 235–271. [Online]. Available: https://doi.org/10.1007/978-1-84882-935-0_5

[25] M. Fiaz, A. Mahmood, S. Javed, and S. K. Jung, "Handcrafted and deep trackers: Recent visual object tracking approaches and trends," 2019.

[26] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019.

[27] C. Dicle, O. I. Camps, and M. Sznaier, "The way they move: Tracking multiple targets with similar appearance," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[28] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, "Joint probabilistic data association revisited," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[29] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '11. USA: IEEE Computer Society, 2011, p. 1201–1208. [Online]. Available: https://doi.org/10.1109/CVPR.2011.5995604

[30] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," *2016 IEEE International Conference on Image Processing (ICIP)*, Sep 2016. [Online]. Available: http://dx.doi.org/10.1109/ICIP.2016.7533003

[31] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 03 1960. [Online]. Available: https://doi.org/10.1115/1.3662552

[32] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109

[33] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," 2017.

[34] S. Sun, N. Akhtar, H. Song, A. Mian, and M. Shah, "Deep affinity network for multiple object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 104–119, 2021.

[35] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M.-H. Yang, "Online multi-object tracking with dual matching attention networks," 2019.

[36] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," 2016.

[37] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," *Lecture Notes in Computer Science*, p. 21–37, 2016. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-46448-0_2

[38] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, ser. AAAI'17. AAAI Press, 2017, p. 4225–4232.

[39] G. Brasó and L. Leal-Taixé, "Learning a neural solver for multiple object tracking," 2020.

[40] S. Lee and E. Kim, "Multiple object tracking via feature pyramid siamese networks," *IEEE Access*, vol. 7, pp. 8181–8194, 2019.

[41] A. Ošep, W. Mehner, P. Voigtlaender, and B. Leibe, "Track, then decide: Category-agnostic vision-based multi-object tracking," 2017.

[42] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, "Mots: Multi-object tracking and segmentation," 2019.

[43] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2019. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2019.00103

[44] J. Zhang, S. Zhou, X. Chang, F. Wan, J. Wang, Y. Wu, and D. Huang, "Multiple object tracking by flowing and fusing," 2020.

[45] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," 2020.

[46] Z. He, J. Li, D. Liu, H. He, and D. Barber, "Tracking by animation: Unsupervised learning of multi-object attentive trackers," 2019.

[47] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "Trackformer: Multi-object tracking with transformers," 2021.

[48] M. Miah, J. Pepin, N. Saunier, and G.-A. Bilodeau, "An Empirical Analysis of Visual Features for Multiple Object Tracking in Urban Scenes," in *International Conference on Pattern Recognition (ICPR)*, 2020.

[49] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 886–893 vol. 1.

[50] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[51] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, p. 84–90, May 2017. [Online]. Available: https://doi.org/10.1145/3065386

[52] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014.

[53] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," 2015.

[54] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," 2016.

[55] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," 2014.

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *Lecture Notes in Computer Science*, p. 346–361, 2014. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10578-9_23

[57] R. Girshick, "Fast r-cnn," 2015.

[58] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016. [Online]. Available: https://proceedings.neurips.cc/paper/2016/file/577ef1154f3240ad5b9b413aa7346a1e-Paper.pdf

[59] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," 2016.

[60] ——, "Yolov3: An incremental improvement," 2018.

[61] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020.

[62] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," 2018.

[63] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," 2019.

[64] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "A self-adjusting approach to change detection based on background word consensus," in *2015 IEEE Winter Conference on Applications of Computer Vision*, 2015, pp. 990–997.

[65] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Image Analysis*, J. Bigun and T. Gustavsson, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 363–370.

[66] D. Du, Y. Qi, H. Yu, Y. Yang, K. Duan, G. Li, W. Zhang, Q. Huang, and Q. Tian, "The unmanned aerial vehicle benchmark: Object detection and tracking," 2018.

[67] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[68] X. Liu, W. Liu, T. Mei, and H. Ma, "A deep learning-based approach to progressive vehicle re-identification for urban surveillance," in *ECCV*, 2016.

[69] L. Yang, P. Luo, C. C. Loy, and X. Tang, "A large-scale car dataset for fine-grained categorization and verification," 2015.

[70] J. Sochor, J. Spanhel, and A. Herout, "Boxcars: Improving fine-grained recognition of vehicles using 3-d bounding boxes in traffic surveillance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, p. 97–108, Jan 2019. [Online]. Available: http://dx.doi.org/10.1109/TITS.2018.2799228

[71] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2015. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2015.7298682

[72] Anaconda, Inc. (2018) Python numba. [Online]. Available: http://numba.pydata.org/

[73] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[74] Z. Kalal, K. Mikolajczyk, and J. Matas, "Forward-backward error: Automatic detection of tracking failures," in *Proceedings of the 2010 20th International Conference on Pattern Recognition*, ser. ICPR '10. USA: IEEE Computer Society, 2010, p. 2756–2759. [Online]. Available: https://doi.org/10.1109/ICPR.2010.675

[75] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "Hota: A higher order metric for evaluating multi-object tracking," *International Journal of Computer Vision*, vol. 129, no. 2, p. 548–578, Oct 2020. [Online]. Available: http://dx.doi.org/10.1007/s11263-020-01375-2