Department of Informatics
Technical University of Munich

Master's Thesis in Informatics

Multi-Vehicle Detection and Tracking in Aerial Imagery Sequences using Deep Learning Algorithms

„Deep-Learning"-basierte Detektion und Verfolgung von Fahrzeugen in Luftbildsequenzen

Supervisor Prof. Dr.-Ing. habil. Alois C. Knoll

Advisor Emeç Erçelik, M.Sc.

Author Tsuyoshi Beheim

Date November 15, 2021 in Garching

Disclaimer

I confirm that this Master's Thesis is my own work and I have documented all sources and material used.

Garching, November 15, 2021 (Tsuyoshi Beheim)

Acknowledgements

I would like to thank Emeç Erçelik, Seyed Majid Azimi, and Reza Bahmanyar for giving me the opportunity to work on this exciting topic. In addition, I am grateful for all their helpful feedback and ideas. I would also like to thank DLR for providing the resources needed for my work. Finally, I would like to express my gratitude to my family and girlfriend for their continuous support.

Abstract

Multi-Object Tracking (MOT) describes the task of identifying all objects in an image and following them over a sequence of frames. A wide range of Deep Learning-based MOT methods has been developed for the autonomous driving domain, but not for aerial imagery. Applications, for example traffic analysis, depend on such tracking capability. This thesis addresses multi-vehicle tracking in top-down aerial imagery sequences. MOT algorithms of other domains are applied on an aerial dataset to create a benchmark. The best method is then identified by using a set of appropriate metrics. Based on that, a variety of modifications are performed to inspect the effect on the tracking capability. This variety consists of a vehicle orientation prediction, an object re-identification feature, and a motion prediction module. Experiments with different configurations are conducted and evaluated. A MOTA score of 78.5 is achieved on the KIT AIS vehicle dataset.

Zusammenfassung

Multi-Object Tracking (MOT) beschreibt das Problem, alle Objekte auf einem Bild zu identifizieren und über eine Sequenz von Bildern zu verfolgen. Viele auf Deep Learning basierende MOT-Methoden wurden für das autonome Fahren entwickelt, aber nicht für Luftaufnahmen. Anwendungen, z.B. die Verkehrsanalyse, benötigen diese Tracking-Fähigkeit. In dieser Masterarbeit wird die Verfolgung von Fahrzeugen auf Luftbildsequenzen thematisiert. MOT-Algorithmen anderer Domänen werden auf einem Luftbilddatensatz angewandt, um einen Benchmark zu erstellen. Die beste Methode wird durch verschiedene Metriken identifiziert. Darauf basierend werden eine Reihe von Modifikationen durchgeführt, um ihre Auswirkungen auf die Tracking-Fähigkeit zu untersuchen. Diese setzen sich zusammen aus einer Vorhersage der Fahrzeugausrichtung, einer Objekt-Wiedererkennungsfunktion und einem Modul für die Bewegungsvorhersage. Es werden Experimente mit verschiedenen Konfigurationen durchgeüfhrt und ausgewertet. Ein MOTA-Wert von 78.5 wird auf dem KIT AIS vehicle Datensatz erzielt.

Contents

Chapter 1

Introduction

In recent years, Deep Learning has revolutionized the way many computer vision tasks are solved [KSH12]. Image classification [KSH12] and object detection [Ren+15] are now tackled using neural networks - a model capable of learning complex relationships within data. Multi-Object Tracking (MOT) can be described as the problem of detecting, identifying and following all objects in a sequence of images [Luo+20]. This is often associated with the field of autonomous driving where pedestrians and other vehicles have to be tracked in order to provide safe decision making for self-driving cars. With many companies developing autonomous vehicles [HNS19] there has been consequently much efforts done in algorithm design and data acquisition. It is however not the sole area in which MOT is required.

Aerial imagery offers new possibilities in solving vision problems by observing events occurring on the ground. Traffic jams, natural disasters and large crowds must be viewed from the sky to be handled meaningfully [21c]. With MOT vehicle trajectories can be analyzed and frequent congestion spots can be located. In case of catastrophes, such as floods tracking people and vehicles provides valuable insights on their location and the water movement. At big events, e.g. concerts, festivals or protests the crowd behavior can be observed to deal with or even prevent emergencies, e.g. mass panic.

This thesis deals with multi-vehicle tracking in aerial imagery sequences. Compared to the field of autonomous driving there is however not much work done which can be attributed to the lack of publicly available datasets and its lower publicity. Algorithms thus are developed primarily for people and vehicle tracking in pedestrian and autonomous driving scenarios. Making use of such methods does not immediately translate to good performance in the aerial imagery domain. This becomes apparent when comparing images of different MOT datasets.

1.1 Differences in Multi-Object Tracking Domains

Figure 1.1 shows a frame that represents a sequence captured by a) a helicopter, b) a car-mounted camera, and c) a low-flying drone. All frames display vehicles, but the perspectives cause different appearances. Frame a) shows a view captured by a camera aimed orthogonally to the ground, in short *top-down view*, also known as *nadir*. Each vehicle is therefore small and presented by the appearance of its roof which shows only minor differences among the other instances. In contrast, b) shows a much larger (side-) view of the cars with one or two of their sides being visible simultaneously. Car lights, license plates, and windows allow for an easier distinction. However, more occlusions can occur, i.e. cars might be visible only

(a) KIT AIS Vehicle [21a]

(b) KITTI [GLU12]

(c) VisDrone [Zhu+20]

Figure 1.1: Different Domains for Multi-Object Tracking

partially, or in some occasions not at all. Frame c) can be described as a mix of the other styles with smaller object sizes, while presenting several sides of a vehicle.

Another difference between the datasets is the frame rate. It can be lower in aerial imagery sequences, e.g. 2 fps vs. 15-30 fps. This causes larger object offsets between frames with usually small or non-existing overlaps. In real-world scenarios, the helicopter is usually not hovering completely motionless above the ground. This results in still objects "moving" from one frame to another within the image. This property, especially coupled with low frame rates, makes vehicle association in consecutive frames quite difficult. In addition, the detection of objects themselves becomes challenging because of the small size, but also because of the large number of vehicles.

1.2 Motivation

To be able to address the tasks mentioned in the beginning, e.g. traffic analysis, aerial images in nadir perspective are considered. This thesis therefore tackles the challenges of multi-vehicle tracking in top-down aerial imagery sequences. To this end, a thorough literature review of MOT methods is performed first. Next, a selection of algorithms is made, benchmarked, and analyzed on the KIT AIS dataset [21a]. The best performing algorithm is identified by using a set of metrics. This algorithm then serves as the basis for a variety of modifications. Experiments are conducted to inspect their effect on the tracking ability. Finally, the results are discussed and a conclusion is drawn that presents possible directions for future work.

This base-algorithm uses a Transformer [Vas+17] architecture to tackle the tracking task. Compared to the other tested methods, it is generally able to detect small vehicles and to follow them even with occasional loss of detection. However, vehicle identities are not always correctly assigned but sometimes confused with each another (identity switch). This issue occurs also with objects that are not vehicles (false positives). To improve these issues, three modifications are implemented:

1. The use of vehicle orientation may help to stabilize the training and inference by constraining inter-frame orientation deviation. Thus, multiple architecture adaptations are performed for vehicle angle prediction.

2. Identity switches can be observed with objects that are visually not similar to each other.[1] For that, a vehicle re-identification function is integrated.

3. The tracking ability may be mostly attributed to a good detection ability, while the association capability of objects between frames may not be of particular strength. Here, a trajectory prediction module is used to replace the existing tracking functionality.

[1] The visual explanation for identity switches will be provided in Subsection 4.2.

Chapter 2

Related Work

2.1 Fundamentals

2.1.1 Deep Artificial Neural Networks

A *Neural Network* (NN) can be seen as a function approximator mapping a k-dimensional input vector to an output vector. They are sometimes called Artificial Neural Networks to be separated from the biological neuron structures in the human brain. It contains a set of connected neurons or "nodes" which transform the given data in a non-linear way. Each neuron takes a linear combination of the input and the parameters, and "activates" the result using a non-linear function. Depending on the data this can be, e.g. the sigmoid function:

$$\sigma(x) = \frac{1}{1+\exp(-x)} \tag{2.1}$$

which maps negative values for x to $[0, 0.5[$ and positive values to $]0.5, 1]$. It is therefore also suited for binary classification problems. This corresponds to a one-layer neural network. If this output is used again as the input to a set of neurons, the network has a second layer. Thus, neurons are structured into layers. A NN is called "deep" if it has at least one *hidden layer*. This hidden layer transforms the input data and passes it to the next layer. The values of the previous layer's neurons are each multiplied with weights in the following way [Bis06]:

$$z_j = h\left(\sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) \tag{2.2}$$

It describes a linear combination of the input data with the weights. The result is activated using an activation function. Depending on the task that is modeled with the neural network the output can be a one-hot vector - in the case of classification - or real values when performing regression.

2.1.2 Process of Learning a Task

The learning process of neural networks can be exemplified with binary image classification. Given an image of a cat, the network is supposed to decide whether a cat is depicted. Initially, the pixel data is vectorized and fed into the first layer. The weights define the importance of every pixel contributing to the classification decision. The connected neurons are then activated or deactivated using an appropriate function. This sequence is in principle repeated for

the remaining layers. Finally, at the last layer the weights connect to a single neuron which represents the binary classification. The logit is activated using the sigmoid function (see Subsection 2.1.1) and the resulting decision is made. This process is performed for every image in the *training* dataset for one *epoch* to be completed. A corresponding *loss function* determines the error between the predicted labels and their ground-truth. An optimizer, e.g. gradient-descent calculates a better guess for the weights of all connections in the network. The learning procedure continues with further epochs until a satisfiable performance is reached. The evaluation is usually measured on an unseen *test* dataset.

2.2 Neural Network Architectures

2.2.1 Convolutional Neural Networks

With large and deep networks however, the computational demand becomes problematic. Since each neuron is connected to all neurons in the previous layer, there are n^2 parameters to be learned with two equally sized layers of size n. Both memory and computation requirements quickly increase. Further, with image recognition tasks a powerful computer vision technique is not utilized which is the *convolution* operation. It is a function which calculates the weighted sum of a fixed set of pixel values in the input image in order to extract visual features. The weights are given by the so called *kernel* or *filter*. A popular filter is the Sobel filter [KVB88] which serves as an edge detector.

The Convolutional Neural Network (CNN) [KSH12] solves these problems by combining the convolutional filters with traditional neural networks. Unlike with conventional filters, the weights are not pre-defined but learned. It is therefore the network that decides which features are relevant for solving the task. With CNNs solely the kernel parameters are learned and thus fewer parameters are needed. With m kernels of size k, there are only $m*k^2$ weights needed with usually $k < m$ and $m << n$.

2.2.2 Transformers

Vaswani et al. [Vas+17] proposed the Transformer - an encoder-decoder model consisting entirely of various attention methods. This architecture was designed to tackle problems particularly in language processing, e.g. text translation and has become the basis for many methods in this field [Wol+20]. So far, sequential data was modeled using Recurrent Neural Networks (RNN) or Long-Short-Term-Memory Networks (LSTM) [HS97]. The former has an internal memory module which temporarily saves the result after passing an input element. This output is reused for processing the next element. This way, contextual information is considered. A short-term memory issue arises, i.e. only information from recent input elements are remembered. The LSTM solves this issue by introducing gates which decide whether information is relevant.

The attention mechanism can be understood as weighted sum of the input sequence with setting focus ("attention") on the relevant parts with a learned function to produce the weights. The Scaled Dot-Product Attention is the core function they use in their architecture [Vas+17]:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.3}$$

General attention is described by the authors as a function mapping a query Q together with a key-value pair K, V to an output. In 2.3 the weights V are obtained by the softmax expression. The softmax-function can be seen as a multi-variate version of the sigmoid-function, e.g. used for a multi-class problems (see 2.1.1). Several of these attention operations are performed in parallel. This aggregation is named Multi-Head Attention block and used throughout the Transformer architecture. The information is processed as follows:
Firstly, the encoder receives the input combined with positional encodings. This is to retain the order of the sequential data when feeding into the attention modules. A layer in the encoder consists of a multi-head attention block coupled to a feed-forward network. The former performs self-attention, i.e. applying attention on the queries, keys and values from the output of the previous layer. The encoder consists of six of these layers in total.
The decoder works in a similar manner. However in addition, each decoder layer receives keys and queries from the output of its encoder counter part and performs multi-head attention with the queries from its previous decoder layer. Finally, the decoder output is fed into a feed-forward and softmax layer to produce probabilities for the next item (e.g. word in a sentence).

2.3 Object Detection

Unlike image classification, object detection deals with localizing and classifying multiple objects in an image. It can be structured into one-stage and two stage detection methods. The latter first uses a region proposal method to suggest approximate image areas in which object are present. Subsequently, the object is classified and the bounding box is regressed. One of the first architectures with this paradigm is R-CNN [Gir+14] with many different successors. One-stage detectors directly predict the bounding boxes with its labels and therefore provide lower inference time, e.g. YOLO [Red+16] and SSD [Liu+16].

Faster R-CNN [Ren+15] makes use of a neural network responsible for the region proposals instead of classic methods, e.g. selective search in R-CNN and Fast R-CNN [Gir15]. It also uses non-maximum-suppression (NMS) [Gir+14] which selects the bounding box with the highest intersection-over-union (IOU) with the ground-truth in case of other predicted boxes located over the same target box. Cascade R-CNN [CV18] uses several R-CNN stages which are trained sequentially, i.e. using the output of the first as input for the next. Each cascade uses a bigger IoU-threshold which solves the dilemma of choosing the correct one. YOLO is an one-stage detector which divides the input image into a grid of cells on which boxes with confidence scores are outputted. The class probability of each box is obtained and IoU is used to select the proposed box with the highest overlap.

2.3.1 Deformable DETR

With DETR [Car+20] Carion et al. propose a fully end-to-end architecture for object detection which in particular does not have the need to rely on manually designed aspects, e.g. definition of anchor sizes or post processing steps, e.g. NMS. For that, the authors combine a regular CNN backbone (e.g. ResNet) with a Transformer encoder-decoder architecture coupled with a feed forward network to directly produce a set of bounding box predictions (without first relying on initial guesses). The Transformer encoder receives CNN features together with positional encodings in order to retain sequential order when pushed through the attention modules. Finally, bipartite matching is applied to associate these with the ground

truth. While DETR achieves comparable performance to the non-end-to-end state-of-the-art object detectors it still has its drawbacks. On certain benchmarks, DETR needs 10-20 times more epochs to reach convergence than Faster R-CNN and has difficulties detecting small objects.

Zhu et al. [Zhu+21] find these aspects to be originating from the attention modules in the Transformer architecture which have problems handling image features. Dai et al. [Dai+17] propose deformable convolutions which is a modified version of the convolutions in CNNs. In contrast to the conventional rigid squared sampling region, they introduce offsets learned from previous feature maps which allow for flexible (deformable) convolutional grids with adaptive receptive fields. Zhu et al. makes use of this concept to propose the *Deformable DETR*. It requires significantly less epochs for convergence during training and overall improves the performance with its strength lying in detecting small objects. The main contribution are the *deformable attention modules* which replace the original attention modules in the Transformer. Similar to deformable convolutions, instead of taking all pixels in the feature map as input for the attention module only a few key points located near a reference point are used.

2.4 Multi-Object Tracking

Object Tracking is the problem of detecting one or several objects simultaneously in each frame and associating them across frames to form trajectories. These tasks are called Single-Object Tracking (SOT) and Multi-Object Tracking (MOT), respectively. In contrast to SOT, identities have to be assigned to all different objects in MOT. Tracking them becomes challenging since one detected box can belong to many possible partial trajectories (tracklet). With that, a common problem arises called Identity Switches (IDSW). These occur when two detections often similar in location or appearance get mistaken for one another and their trajectories falsely continue.

2.4.1 Online- vs. Offline Multi-Object Tracking

MOT methods are applied in various scenarios, e.g. autonomous driving, traffic observation and analysis, sports analysis, etc. The importance of tracking capability and speed depend on the actual task that is solved. When analyzing previously captured video data for MOT, speed often plays a less significant role than in autonomous cars where other vehicles and pedestrians have to be registered in real-time. Thus, MOT can be categorized into online- and offline methods. Online MOT uses only the current and a few previous frames - and in particular no data of unseen events - to perform data association. In offline MOT, batches of frames are available and used to provide more data to improve tracking results. Thus, occlusions are simpler to tackle when there is potential access to the object's future location in a visible area (e.g. when a vehicle passes under a bridge).

2.4.2 MOT Paradigms

Tracking-by-Detection

An intuitive way to track objects consists of using a detection model together with a association logic. This is called Tracking-by-Detection which is shown in Figure 2.1. The detector

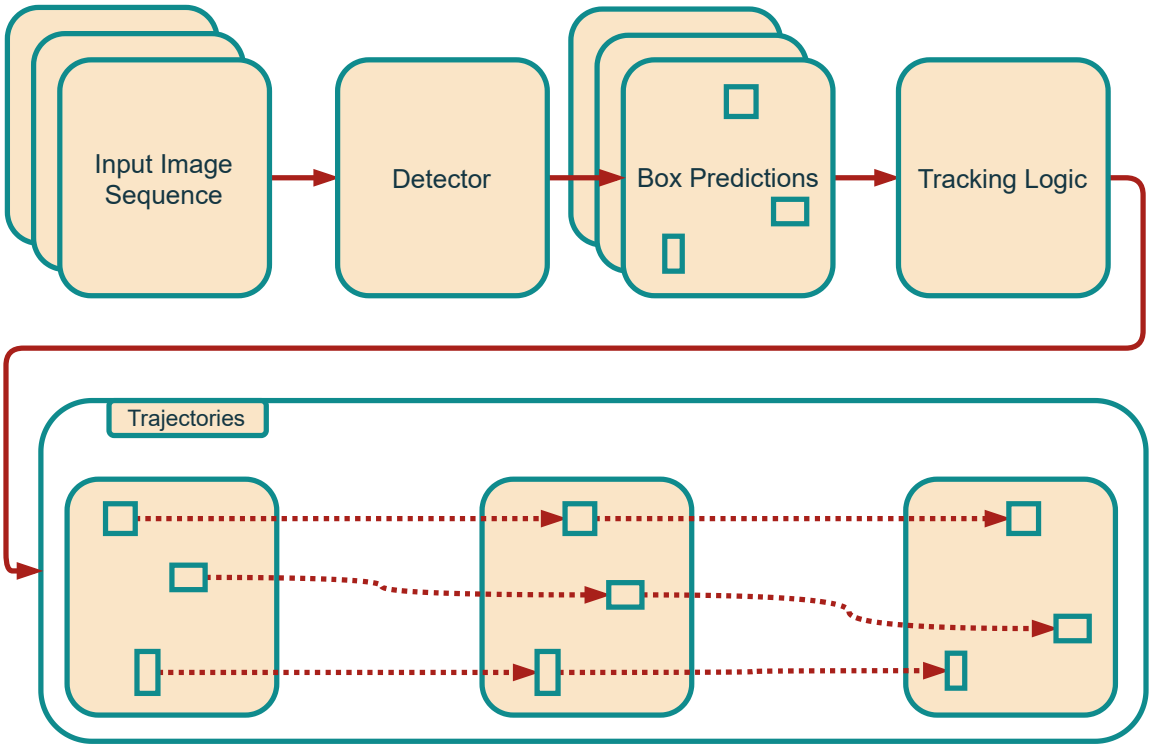


Figure 2.1: Tracking-by-Detection

provides the bounding boxes for all frames which are then linked across the frames to form trajectories in a second step. The advantage of this paradigm is the convenience of picking one of the many strong detectors. With a sufficient high frame rate an object's location usually does not deviate by much between two consecutive frames. Thus, Bochinski et al. [BES17] made use of the overlap (IoU) between objects over time to judge which track to associate with. This method requires a clear sight on the target objects to produce good tracking performance.

Another approach of dealing with the association problem is the trajectory prediction. Here, every object's location in the current frame is predicted using information gathered from past frames. This adds another constraint to linking a possible object to an existing track. One method to accomplish this task is the Kalman Filter [WB06]. It estimate a stochastic process by alternately predicting the state and performing a correction by using measured data at each time-step. For object tracking, this translates to optimizing the balance between location prediction and the actual detected bounding boxes. SORT uses the Kalman Filter to predict the box positions and links them to the current detections with the highest IoU. Neural Network-based approaches have also been used for motion prediction, in particular Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) Networks. Milan et al. [Mil+16c] utilized RNNs for the temporal prediction and LSTMs for the association part. Chaabane et al. [Cha+21] used a LSTM to provide motion constraints.

In addition to using motion information, Wojke et al. [WBP17] extends SORT by extracting features using a CNN to distinguish between objects and thus make the association problem also dependent on appearance information. This is also called (Person) Re-Identification (Re-ID). Often multiple instances of the same class are to be observed, e.g. pedestrians, vehicles, etc. These already have similar appearance features, e.g. the shape or the color, which makes the association task particularly demanding. For that, appearance models can be utilized.

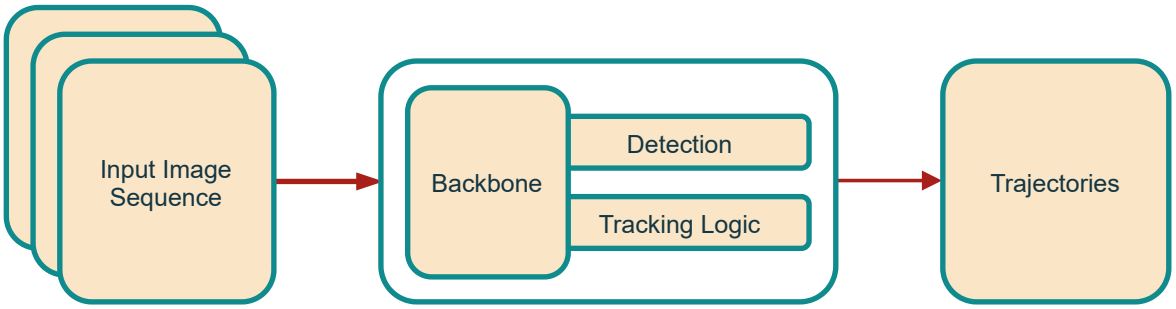


Figure 2.2: Joint-Tracking-and-Detection

They embed the objects visual features in order to more effectively perform similarity comparisons between object candidates for association ([Che+18]).

Siamese Networks (SNN) can be used when comparing pairs of image regions. Both patches are processed in parallel by two network branches sharing architecture and weights. Leal-Taixé et al. incorporated a SNN to learn and assess the similarity between objects for the linking stage. Though not necessarily following this paradigm, SNNs are popular for recent SOT methods [Li+18] [Wan+19], also owing to their fast inference capability [HTS16]. With this paradigm, one can choose and optimize each each component (detection, motion, re-ID) separately.

Joint-Tracking-and-Detection

Despite their popularity, tracking-by-detection algorithms have their drawbacks. Since detection models are used "off-the-shelf", the tracking ability is limited to that of the detector. Further, tracking and detection are handled as two separate tasks and thus, no information is shared between the two modules. Apart from the arising redundancy issue, each component in the algorithm increases the inference time, as well. This is the motivation for the recent Joint-Tracking-and-Detection solutions. The general concept is shown in Figure 2.2. In general, a detector is converted into handling detection and tracking tasks ("Tracking Logic") together. Consequently, such architecture can be trained end-to-end. The same backbone CNN is usually used for both tasks to share features. This way the tracking task is not secondary anymore. Tracktor uses the regression head of a detector to regress the object's current position using the previous location. A re-ID capability has been added in form of an additional branch to the detector [Zha+20] or as an embedding head [Wan+20]. Detailed explanations to other algorithms following this paradigm can be found in Section 3.4.

2.5 Multi-Object Tracking in Aerial Imagery Sequences

The majority of MOT algorithms are developed on data captured by cameras near ground-level. This is done to depict scenarios in autonomous driving or pedestrian zones. A popular dataset for the former is the KITTI dataset [GLU12] which contains traffic sequences recorded from a vehicle-mounted camera. Vehicles are therefore displayed as rather large objects (relative to the frame size). Pedestrian scenarios are captured in the MOT Challenge [Mil+16a] with varying view points. Overall, the sizes of the people are relatively large as well since even the highest camera position is still only a few meters above ground. In contrast to these datasets, aerial images display a much larger area on the ground and thus more and in particular smaller objects. In Object Detection several high quality aerial datasets can be

found. DOTA [Xia+18] contains different classes, e.g. vehicles, planes, ships, soccer fields, etc. while EAGLE [Azi+21] is focused on vehicle detection. When considering aerial MOT, far less datasets are publicly available. VisDrone [Zhu+18] contains high-resolution image sequences with annotations for cars, trucks, pedestrians, etc. However, most of these frames are not captured by a nadir-looking perspective but have a much more narrow angle to the ground. The KIT AIS vehicle dataset [21a] provides sequences captured entirely by nadir-looking cameras for vehicle tracking. As this scenario exactly fits the task of this thesis most this dataset will be used for the benchmark 3 and the experiments 4.3.

SMSOT-CNN [BAR19] is based on GOTURN and uses micro CNNs to perform SOT on each object. It is evaluated on the KIT AIS vehicle dataset and achieves a MOTA of 41.1. The VisDrone 2020 Challenge [Fan+20] presents the top performing approaches on their MOT dataset. Several submissions in the top 10 follow the tracking-by-detection paradigm using the Cascade-RCNN detector [CV18]. Among them is the 1st placing method called "COFE" which uses the Kalman Filter for motion prediction and OSNet [Zho+19] as the re-ID model. PAS Tracker [SSB20] uses a mix of location, appearance and size information as a similarity metric for data association. Many approaches are based on IOU-Tracker. Others include methods using SORT and CenterTrack [ZKK20]. The latter is a point-based tracking method which is described in Subsection 3.4. In general, more algorithm publications can be found in the people tracking and autonomous driving field.

Chapter 3

Benchmarking

To tackle the task of multi-vehicle tracking in aerial imagery, several existing tracking algorithms are put to the test. Time and computational resources are limited, thus criteria for this selection are established. These methods are then benchmarked on the KIT AIS dataset [21a] which is presented in 3.2. The method exhibiting the best performance is identified using appropriate metrics.

3.1 Preliminaries

3.1.1 Selection Criteria

The goal of the benchmarking phase is to find an algorithm which is able to tackle the challenges of vehicle tracking in aerial imagery as best as possible. In particular, this method should perform well on data captured by a top-down birds-eye-view (nadir) perspective. The only MOT dataset containing entirely such image sequences is the KIT AIS dataset. The VisDrone dataset [Zhu+20] includes aerial data which is however mostly comprised of frames depicting a flat-angled BEV perspective (see Figure 1.1).

The algorithms are chosen by inspecting the rankings of the two popular multi-object tracking challenges based on the MOT and KITTI dataset, respectively. Newly developed algorithms are often evaluated on these datasets. This can be seen with the algorithms in Section 3.4 as they were all presented in the last two years. Additionally, nearly all of them are evaluated on the MOT Challenge dataset. Therefore, the ranking list offers a pre-selection of state-of-the-art object tracking algorithms. As explained in Section 3.2, the MOT and KITTI datasets handle multi-pedestrian and vehicle tracking from a frontal car view, respectively. On the other hand, the VisDrone Challenge provides a ranking of algorithms closer to the area of application of this thesis. However, they often do not make use of the latest architectural components which was shown in Section 2.5. Thus, even though it differs from the top-down view scenarios, the algorithms were selected from the MOT and KITTI challenges in hope of benefitting from state-of-the-art concepts. Further constraints for the decision are the existence of a publicly accessible code-base and a corresponding paper.

3.1.2 Benchmarking Conditions

The implementation details are taken from the respective papers. In some cases, various training configurations exist which depend on the specific dataset the algorithm is run on.

For instance, CenterTrack provides configurations for KITTI, CrowdHuman [Sha+18] and MOT Challenge. For consistency reasons, the configuration for the latter is chosen. Since none of the selected algorithms were evaluated on the KIT AIS dataset in their respective papers, the training configurations used for this benchmark might not produce the optimal results for each algorithm. Under these circumstances, the method with the best results is selected.

3.2 Dataset

The KIT AIS Dataset contains nine sequences of frames captured from a nadir perspective. In total there are 239 frames. The resolution and aspect ratio differ but stay consistent within a sequence. The width and height in pixels range from 633 and 377 to 1771 and 988, respectively. It does not contain a publicly accessible official test set. In the literature [Azi+20] [BAR19], a partition of five training and four test sequences can be found. This split was used for this benchmark and the experiments in Chapter 4.3, as well. In particular, the test sequences are MunichCrossroad02 (MC02), MunichStreet (MS02), MunichStreet04 (MS04) and StuttgartCrossroad01 (SC01).

3.3 Metrics

As explained in Section 2.4, the goal of multi-object tracking is the detection and correct identification of each object in every frame. Common shortcomings are loss of detections, false assignment of new identities or mistaking ids of two objects (identity switch). To measure the performance of an MOT algorithm and capture such mistakes as complete as possible, several metrics have been proposed in the literature.

3.3.1 MOTA and MOTP

Bernardin et al. [BS08] established the CLEAR MOT metrics which consists of two metrics. The first one is the Multi Object Tracking Accuracy (MOTA) which is defined as follows:

$$\text{MOTA} = 1 - \frac{\sum_t (\text{FN}_t + \text{FP}_t + \text{IDSW}_t)}{\sum_t \text{GT}_t} \tag{3.1}$$

In this notation by Milan et al. [Mil+16b], the mistakes made by an algorithm regarding tracking and detection are calculated in all frames $t \in T$. FP and FN are false positives and false negatives, respectively. IDSW describe identity switches, i.e. the number of occasions an object x is mistaken with object y. GT_t are the ground truth detections at frame t.
The second metric is the Multi Object Tracking Precision (MOTP):

$$\text{MOTP} = \frac{\sum_{i,t} d_t^i}{\sum_t c_t} \tag{3.2}$$

Here, the error d_t^i occurring between the matching of ground truth and prediction boxes is averaged over all matches c_t actually established by the algorithm. MOTP therefore captures the localization ability of the algorithm and in particular does not express the association quality.

3.3.2 ID metrics

Ristani et al. [Ris+16] consider the identities of trajectories and establish metrics that match the whole ground-truth trajectories with all computed ones. Using this scope, they reformulate conventional metrics, e.g. TP, FP and F1 as IDTP, IDTP IDFP and IDF1. The latter is defined as follows:

$$\mathrm{IDF1} = \frac{2\,\mathrm{IDTP}}{2\,\mathrm{IDTP} + \mathrm{IDFP} + \mathrm{IDFN}} \tag{3.3}$$

For this benchmark, the metrics in particular mentioned in this Section (3.2) are utilized to analyze the performance of all algorithms. All metrics stated in Table 3.1 [21b] contribute to assess the performance of an algorithm. However, in order to capture the association capability, MOTA is best suited and thus used as the primary comparison metric.

Metric	Description
IDF1	The F1 score based on the ID Metrics, see Subsection 3.3.1
IDP	The ID Precision
IDR	The ID Recall
Rcll	The number of TP detections over all ground-truth boxes
Prcn	The number of TP detections over all detected boxes
FAR	False Alarm Ratio
GT	Ground-Truth identities
MT	Mostly Tracked: All trajectories with >80% target matching rate
PT	Potentially Tracked: All trajectories with between 20% and 80% target matching rate
ML	Mostly Lost: All trajectories with <20% target matching rate
FP	False Positives
FN	False Negatives
IDs	Identity Switch
FM	Fragmentations
MOTA	Multi-Object Tracking Accuracy, see Subsection 3.3
MOTP	Multi-Object Tracking Precision, see Subsection 3.3
MOTAL	MOTA but with log(IDs)
AE	Angle Error, see 5.2.1

Table 3.1: Metrics for Benchmark and Experiments

3.4 Algorithm Selection

The following algorithms are selected using the aforementioned criteria.

3.4.1 CenterTrack

In "Tracking Objects as Points" [ZKK20] by Zhou et. al each bounding box is represented as its center coordinates and tracked over each frame. The middle point is found using CenterNet

[Dua+19] as the detector backbone. It takes the current and previous frame as an input together with a heatmap of points describing the tracklets. With each outputted detection, an offset vector to the center of the object in the previous frame is generated to follow motion locally between two frames. The predicted offset is used to associate each current detection with an unmatched box of the previous frame.

3.4.2 DEFT

"Detection Embeddings for Tracking" [Cha+21] by Chaabane et al. makes use of extracted appearance features from the detector in multiple scales. These object embeddings are used in the matching network to associate objects in previous frames with current detections. This sub-network is trained jointly with the detector. This improves the matching features which would otherwise depend completely on those learned by the detector that have a different purpose. In order to avoid unrealistic associations, i.e. large object displacements between consecutive frames, a LSTM predicts an objects motion and thereby sets constraints on its future location.

3.4.3 FairMOT

In this work by Zhang et al. [Zha+20], the unfairness caused by treating object detection and re-ID as two independent tasks is examined and solved. A network architecture with two parallel branches for detection and re-ID is proposed. The latter uses a classification approach to describe its loss, i.e. every object in a frame is assigned to a separate class which holds all instances with the same id. Similar to DEFT, the objects motion is predicted to filter out improbable associations during inference. For that, a Kalman Filter is used.

3.4.4 TraDeS

"Track To Detect and Segment: An Online Multi-Object Tracker" [Wu+21] by Wu et al. proposes a joint-detection-and-tracking architecture that uses tracking information to improve object detection ability. In particular, they propose a cost volume based approach to tackle the re-ID problem and a module which uses motion information to propagate previous object features to the current frame.

3.4.5 CSTrack

As the title suggests, "Rethinking the competition between detection and reid in multi-object tracking" [Lia+21] by Liang et al. deals with the effect of detection and re-ID part of a network on the MOT ability. Similar to JDE, two branches are incorporated to handle detection and re-ID tasks. Further, a cross-correlation network is proposed which avoid the competence between both tasks. While DEFT solely extracts feature embeddings from layers of different scales for their matching head CSTrack utilizes a scale-aware attention network. This network applies spatial and channel-wise attention mechanisms on different-scaled input features to create fused embeddings useful for the re-ID task.

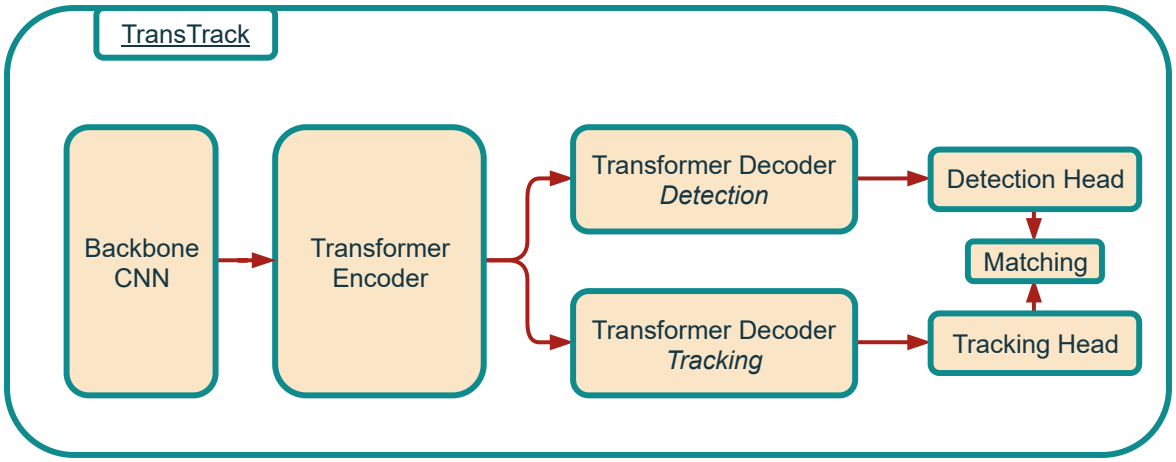


Figure 3.1: The Pipeline of TransTrack

3.4.6 TransTrack

Sun et al. [Sun+21] are allegedly the first to propose a Transformer-based MOT method. Its architecture is inspired by the Deformable DETR (see Subsection 2.3.1) which is a Transformer-based object detector. The pipeline is depicted in Figure 3.1. It is helpful to see that the backbone, encoder, detection decoder and detection head constitute the parts forming the Deformable DETR.

The CNN backbone first generates feature maps of two consecutive frames to capture sequential information. These are then fed into the encoder. The encoder outputs feature maps which are used as the key and value by both decoders. The first decoder performs object detection by using the learned object query to find all objects in the frame. The predictions are matched with the ground-truth objects which makes the detection branch similar to the Deformable DETR. On the other hand, the tracking decoder takes care of predicting the objects in the current frame. For that, it uses the track query which are the features obtained from the detection branch at the previous frame. This way appearance and location information are used for the prediction. Both decoders run simultaneously which makes TransTrack a joint-tracking-and-detection method. They output object and track features which are fed into the detection and tracking head to produce the two types of bounding boxes. Finally, these boxes are matched by applying the Hungarian Algorithm on their IoU similarity.

3.5 Benchmarking Results

ALGO	IDF1 ↑	IDP ↑	IDR ↑	Rcll ↑	Prcn ↑	FAR ↓	MT ↑	PT ↑	ML ↓	FP ↓	FN ↓	IDs ↓	FM↓	MOTA ↑	MOTP ↑	MOTAL ↑
FairMOT	0.8	1.2	0.6	0.8	1.6	21.6	0	1	229	2330	4936	9	9	-46.3	57.7	-46.1
CSTrack	8.6	12.6	6.6	13.1	25	18.1	6	36	188	1949	4325	216	157	-30.5	60.4	-26.2
TraDeS	0	0	0	0.4	5	3.5	0	1	229	380	4954	0	0	-7.2	61.4	-7.2
DEFT Kalman	57.2	58.3	56.1	69.7	72.4	12.3	120	63	47	1324	1507	485	217	33.3	70.6	43
DEFT LSTM	46.4	47.3	45.6	72.5	75.3	11	129	60	41	1184	1367	718	175	34.3	72.8	48.7
CenterTrack	69.5	66.5	72.9	84.6	77.1	11.6	173	36	21	1251	768	191	134	55.6	74.7	59.4
TransTrack	86.8	83.7	90.1	93.3	86.6	6.6	205	19	6	716	333	41	85	78.1	81.3	78.9

Table 3.2: Total results of the algorithms on the KIT AIS test set.

The results of each algorithm summed up over all test sequences is presented in Table 3.2.

The order is defined by ascending total MOTA. What stands out is that three entries show negative MOTA values (FairMOT, CSTrack and TraDeS) while that of the others are positive (DEFT, CenterTrack and TransTrack). Values below 0 are valid and occur when the total number of mistakes (as described in Subsection 3.1) reach a certain high amount. In this case, the first three algorithms all suffer from a high amount of false positive and false negative detections. So, for these to be able to tackle the aerial tracking task, a thorough adjustment of training hyper parameters and possibly other architectural properties would have to be performed. It is recalled that all evaluated methods have not been tested on the KIT AIS vehicle dataset so far (to the best of my knowledge) and in particular are not designed for aerial object tracking but people and vehicle tracking on ground-level (see Subsection 3.1). When designing an algorithm, a subset of the training dataset is used for validating the performance. Thus, they are in general less likely to work on other data than, in this case, MOT or KITTI. This phenomenon is also observed in the VisDrone MOT Challenge [Fan+20]. A method based on FairMOT is presented which achieves very low performance on the aerial MOT dataset. As for architectural adjustments, CSTrack for example uses a YOLO detector as its detection backbone which requires previously set parameters, i.e. anchor box sizes that are dependent on the objects it has to detect. The chances of having success with a method are higher if it performs well "out-of-the-box". Therefore, the first three entries are not considered.

3.5.1 Deft

SEQ	IDF1 ↑	IDP ↑	IDR ↑	Rcll ↑	Prcn ↑	FAR ↓	GT	MT ↑	PT ↑	ML ↓	FP ↓	FN ↓	IDs ↓	FM↓	MOTA ↑	MOTP ↑	MOTAL ↑
MC02	39.3	42.7	36.3	63.7	74.8	10.3	66	23	26	17	462	783	243	93	31	71.1	42.1
MS02	50.7	49.1	52.4	69.6	65.1	13.9	47	31	9	7	278	227	102	26	18.6	74.2	32
MS04	53.3	50.4	56.6	90.5	80.6	11.4	68	58	8	2	331	144	293	29	49.4	73.8	68.6
SC01	46	51.1	41.9	61.6	75.1	8.1	49	17	17	15	113	213	80	27	26.7	72.9	40.8
SUM	46.4	47.3	45.6	72.5	75.3	11	230	129	60	41	1184	1367	718	175	34.3	72.8	48.7

Table 3.3: Results of DEFT with a LSTM on the KIT AIS test set.

Deft is evaluated in two configurations: With Kalman Filter and with a LSTM as the motion model. The detailed results for the latter can be seen in Table 3.3. Both have similar MOTA scores with the LSTM version performing better by 1 point (34.3). This follows from less FP and FN but more IDSW which results in worse ID ratings. Overall, 718 IDSW can be seen which are too many for DEFT to be considered suitable for the aerial MOT task.

3.5.2 CenterTrack

SEQ	IDF1 ↑	IDP ↑	IDR ↑	Rcll ↑	Prcn ↑	FAR ↓	GT	MT ↑	PT ↑	ML ↓	FP ↓	FN ↓	IDs ↓	FM↓	MOTA ↑	MOTP ↑	MOTAL ↑
MC02	64.4	58.2	72	85.5	69.1	18.3	66	50	12	4	824	312	67	64	44.2	73.8	47.2
MS02	53.3	63.7	45.8	62.5	86.8	3.6	47	26	9	12	71	280	66	23	44.1	75.4	52.7
MS04	82.9	79.9	86	94.4	87.7	7	68	61	6	1	202	85	38	22	78.6	76	81
SC01	72.6	68.9	76.7	83.6	75	11	49	36	9	4	154	91	20	25	52.2	73	55.5
SUM	69.5	66.5	72.9	84.6	77.1	11.6	230	173	36	21	1251	768	191	134	55.6	74.7	59.4

Table 3.4: Results of CenterTrack on the KIT AIS test set.

CenterTrack tops the results of DEFT in nearly all metrics. Its results are shown in Table 3.4. With a MOTA of 55.6 it at least partially tracks 209 of 230 identities and has much

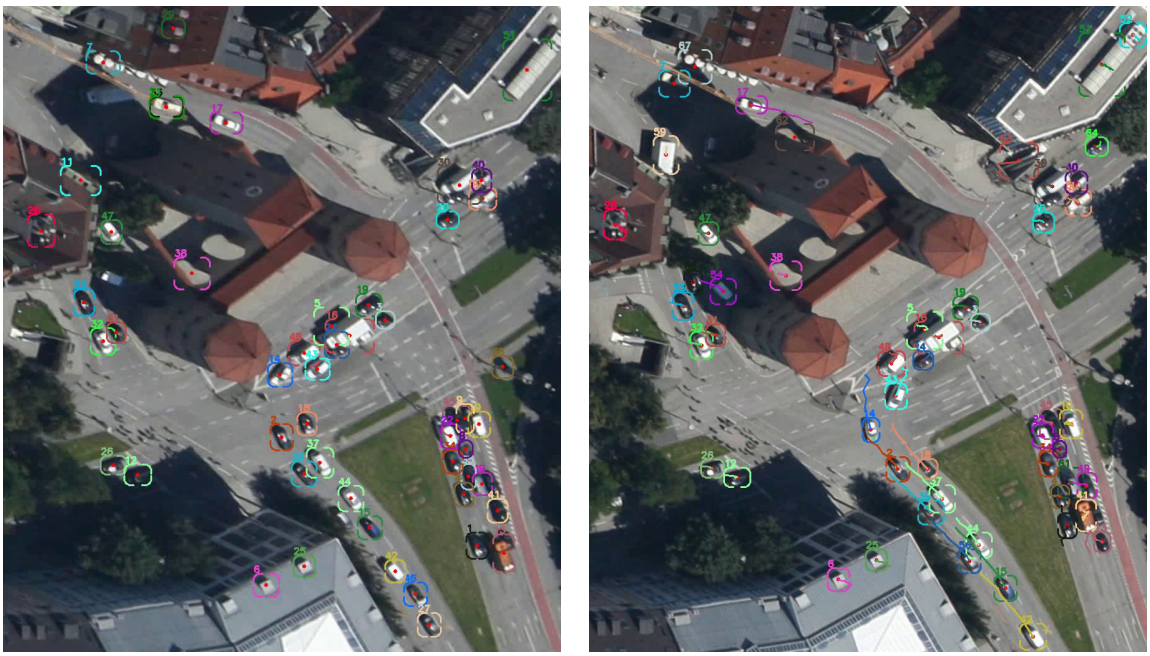
less IDSW (191). The CenterNet detector also performs well with high recall and precision scores. However, the relatively low IDSW count of 191 shows that the association capability is its strength.

3.5.3 TransTrack Evaluation

SEQ	IDF1 ↑	IDP ↑	IDR ↑	Rcll ↑	Prcn ↑	FAR ↓	GT	MT ↑	PT ↑	ML ↓	FP ↓	FN ↓	IDs ↓	FM↓	MOTA ↑	MOTP ↑	MOTAL ↑
MC02	82.1	77.5	87.2	90.8	80.6	10.4	66	56	8	2	470	198	10	44	68.5	82.3	69
MS02	92.8	92	93.7	94.2	92.5	2.9	47	44	2	1	57	43	2	11	86.3	79.7	86.5
MS04	91.6	89.6	93.6	97.1	92.9	3.9	68	63	5	0	112	44	24	23	88.2	81.8	89.6
SC01	84.6	82.5	86.8	91.3	86.8	5.5	49	42	4	3	77	48	5	7	76.5	78.4	77.3
SUM	86.8	83.7	90.1	93.3	86.6	6.6	230	205	19	6	716	333	41	85	78.1	81.3	78.9

Table 3.5: Results of TransTrack on the KIT AIS test set.

TransTrack outperforms CenterTrack in every metric. The far less FP and FN scores and only 41 IDSW produce the best MOTA in this benchmark with 78.1. With this score, it surpasses the performance of SMSOT-CNN which achieves 41.1.
In Figure 3.2, qualitative results can be seen. The first frame shows all initial detections. The right image shows the 10th frame with trajectories attached to the bounding boxes. The colored line is made to be only shown as long as the object is continuously tracked. For example, the orange ID 18 in the middle is correctly followed from start to finish because of the trajectory having the full length. On the other hand, the light-pink truck with ID 59 can be seen only in the right figure. The trajectory is relatively short and has only been tracked for a few frames. This would be an instance captured in the partially tracked PT value which is listed in Table 3.5 in the entry for MC02. Overall, many true positive detections can be seen with mostly correctly tracked instances. On the top of the roofs some vehicle-like shapes are being detected which explains the FP value.



(a) Frame 1 **(b)** Frame 10

Figure 3.2: Sequence MC02 produced by *TransTrack*

Chapter 4

Methodology

The results of Chapter 3 showed that TransTrack has the most promising performance for MOT in top-down aerial vehicle tracking. Thus, this algorithm served as baseline for the experiments. The tracking ability was investigated in regards to three modifications: Vehicle orientation prediction using angle information, vehicle re-identification (Re-ID) and motion prediction.

4.1 Angle Incorporation

With the goal of improving tracking in aerial imagery sequences the orientation of vehicles is predicted. The KIT AIS dataset contains information for horizontal bounding boxes (HBB) and oriented bounding boxes (OBB). The former describes rectangular boxes with axis-parallel sides. The latter defines a rectangular box which encloses the object. In this dataset, the OBB is expressed by an angle which rotates box coordinates for an exact fit. OBB regression is often implemented using five or eight parameters [YYH20]. The first approach regresses (c_x, c_y, w, h, θ) which are the center coordinates, width and height of the object together with the angle [Yan+18]. The second approach does not regress the angle but directly the corner coordinates $(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$.

With properly predicted vehicle orientation, the tracking may be made more robust by comparing angles of an object between two frames. The absolute difference between this pair of angles is called *angle deviation*. Even in low frame rate sequences, the assumption is the angle deviation remains small, i.e. a car does not flip by 45° from one frame to another. For these methods, experiments with OBBs are conducted. In this dataset, most deviations lie within 10°. From this it follows that a precise angle prediction is required.

4.1.1 Angle Representation

In order to correctly predict the orientation of a vehicle there are a few things to consider. In bounding box regression, the coordinates locating the box are predicted, e.g. top-left and bottom-right or the center point with box width and height. The absolute difference described by the L1 loss of a pair of ground-truth and prediction coordinates appropriately describes the resulting error. However, when dealing with angles, such loss function would not represent the deviation correctly when dealing with degree values. In order to illustrate this problem, a target car and its prediction with an angle of 1° and 359°, respectively are considered. Both are similarly oriented. The network however would perceive their direction as quite different

and punish the prediction with a high loss of 358. Thus, it is important to incorporate the periodicity. The angle a is therefore expressed in radians and located uniquely on the unit-circle using $a = cos(\theta), sin(\theta)$. This approach has also been used in [HVC17] and [Don+20]. During inference time, the predicted angle can be obtained by the atan2-function [21d] which returns the angle in radians w.r.t to the (1,0) vector as known from the unit-circle. This way the boundary problem between 0 and 360 degrees is counteracted.

4.1.2 Angle Regression

With the representation described in 4.1, the angle regression can be incorporated into the network architecture. First, the ground-truth angles are converted to cosine and sine values. On top of the decoders and simultaneously in parallel to the bounding box head, the angle regression head is placed. It consists of two linear layers with an output dimension of 2. These are coupled to a smooth-L1 loss function [Gir15]. This loss is a variant of the L1 loss but outputs quadratic element-wise errors if these errors are small. With this formulation, the loss of both cosine and sine values are captured. The bounding box representation of (c_x, c_y, w, h) can remain since coupled with a it represents the five-parameter method.

4.1.3 Angle Classification

Even with a proper angle representation, relying only on regression may not be sufficient for orientation prediction. Providing a range of values the prediction must lie in can lead to improved orientation detection [ZDW20] [HVC17]. Intuitively, the prediction is thereby steered into the approximate correct direction.

Thus, the effect of angle classification as an auxiliary head is examined. The ground-truth for this task has to be reformulated. The range $[0, 360]$ is divided into 16 *sectors* with each sector covering a 22.5° range. With that, the loss does not punish predicted angles that are slightly off. A 16-dimensional one-hot-vector defines the sector the angle of the vehicle belongs to. This serves as the ground-truth representation. Similarly to the angle regression, an angle classification head is implemented. Located next to the other heads, it consists of the linear layer Loutputting a 16 dimensional vector. The loss is then determined by the Cross-Entropy Loss [Nie15]:

$$CE(s) = -\frac{1}{n} \sum_{x} \sum_{j \in [16]} \left[y_j \ln a_j^L + \left(1 - y_j\right) \ln\left(1 - a_j^L\right) \right] \tag{4.1}$$

Some angles, or to be exact, angle sectors may appear more frequently in the data than others. The Cross-Entropy Loss does not account for such imbalance. Lin et al. [Lin+17] tackle this class imbalance issue in object detection by proposing a modification of the Cross Entropy loss called Focal Loss. In one-stage object detectors discussed in Section 2.3 the majority of anchors produce bounding boxes belonging to the background class - meaning no object is present. Compared to the actual boxes of interest, the ratio can be several folds higher. Even with anchor-free methods this problem persists as described in [Che+20]. The Focal Loss considers distinguishing between easy and hard examples, i.e. examples easily or hardly classified as correct. The author introduce a modulating factor which controls the impact on the loss when dealing with easy/hard examples. The loss is calculated as [Lin+17]

$$\mathrm{FL}(p_\mathrm{t}) = -\alpha_\mathrm{t} (1 - p_\mathrm{t})^\gamma \log(p_\mathrm{t}) \tag{4.2}$$

(a) Frame 1 of MC02 (crop)

(b) Frame 10 of MC02 (crop)

Figure 4.1: IDSW occurrence with TransTrack

with $\mathrm{CE}(p_t) = -\log(p_t)$ defining the relation to the Cross-Entropy.

4.2 Re-ID

4.2.1 Identity Switches

As described in Section 2.4.2, one way to assist the association of detections to tracklets is to incorporate a re-identification logic. A stricter appearance model can help to prevent IDSW when two cars are located close together but differ in their looks. Colors and shapes are intuitive features to identify and distinguish objects. This problem indeed occurs with TransTrack which can be seen in Figure 4.1. As explained in Chapter 3, the colored lines show the trajectories and help to estimate whether the tracking has been done correctly over the past frames. Considering the large orange vehicle in each frame the issue becomes apparent. On the left, this vehicle has the ID 13 while on the right it is assigned the ID 41. This is exactly the scenario which is covered by the IDSW in the definition by [Lui+20]. If the vehicle would be drive alone on the road this issue would not happen. The close proximity of cars led to this mistake. In general, it can be said that each drawn trajectory which has been lost indicates the occurrence of an IDSW unless the object receives the same ID after reappearance. Another closely related issue can be seen with ID 13 "jumping" from the orange to the black vehicle from the left to the right frame. This is called ID Transfer. [Lui+20].

4.2.2 Re-ID Losses

In fact, Sun et al. [Sun+21] have presented in their ablation study a TransTrack version with a Re-ID head and branch, respectively. They reported worse performance for both methods with the re-ID branch being closer to the baseline. However, they do not provide any infor-

mation on implementation details. Special loss functions in this context help to improve the tracking performance. They are categorized into (similarity) metric losses, e.g. triplet loss [HBL17] and contrastive loss [SWT15] and classification losses, e.g. L2-softmax [RCC17], CosFace [Wan+18], Angular Loss [Wan+17] and circle loss [Sun+20]. The former optimize on the similarity between object embeddings while the latter assign each sample identity to the same group and treat it as a classification problem. Classification losses have been used in MOT, e.g. FairMOT (see Subsection 3.4.2) uses a softmax loss.

Circle Loss

The circle loss shows a superior performance compared to other classification-based loss solutions, including a 5% lead on face recognition tasks compared to the softmax. It offers a unified formula for both metric and classification loss. The goal is to maximize the intra-class similarity s_p (Goal I) while minimizing inter-class similarity s_n (Goal II). Previous Re-ID loss approaches, e.g. triplet loss try to minimize $\left(s_n - s_p\right)$. In this formulation, both goals have the same impact, i.e. the outcome is the same whether only Goal I is pursued or Goal II. A penalization would effect both s_p and s_n equally which is not always optimal. In this context *positive* and *negative* describe samples that belong to a class or not. An *anchor* is the respective reference object. When positive samples are too far away from the anchor then the emphasis should lie on optimizing Goal I, and vice versa. This leads simply to the following weighted formulation:

$$\left(\alpha_n s_n - \alpha_p s_p\right) \tag{4.3}$$

The name of this loss becomes clear when considering its decision boundary $\alpha_n s_n - \alpha_p s_p = m$ which has the shape of a circle. Further details can be found in their paper [Sun+20].

4.2.3 Re-ID Branch

With these finding a re-ID branch similar to the one presented in the ablation study [Sun+21] is built into the architecture. An independent branch is also encouraged by FairMOT in order to learn task-related features separately. While in the baseline the object queries and the key-value pairs are fed into the decoder, a second decoder is incorporated which acts as the re-ID branch and receives its own set of queries and key-values. Similar to usual classification, the head consists of a linear layer producing a 512-dimensional output. The features of the metric head act as the input for the reid-classification head. This classification vector is used together with the target IDs of the vehicles in the frame to calculate the circle loss. The number of target IDs is of course usually much lower than 512. While training, only the circle loss is responsible for optimizing the re-ID loss. During inference the re-ID features from the metric head are fetched. Using the cosine similarity the cost between detection and tracking box features are assessed. This cost together with the IoU cost between detection and tracking boxes contribute to the final matching decision. It is important to see that the re-ID functionality gives additional input and does not replace the previous cost decision from the baseline.

4.3 Motion Prediction

To remind, the baseline has a detection and a tracking branch implemented as Transformer decoders. The feature map outputted by the detection decoder is then used during the next

frame as the track query to predict the location of the objects. During inference, detection and tracking boxes are matched. Unmatched tracking boxes are saved for a number of frames to be able to be re-matched once their detection is "visible" again. This is also known as track-rebirth [ZKK20]. Qualitatively, in Figure 3.2 long trajectories can be seen that suggest good prediction ability. Quantitatively, the high MOTA and (MT) scores in Table 3.5 indicate great tracking performance, as well. In order to analyze and relate its prediction performance, the predicted tracking boxes during inference are replaced with a LSTM-based motion model. This model outputs the future location given an object's location and other inter-frame properties. As DEFT showed an improvement with its motion model in Section 3.5 its training is used with adapting the architecture for creating two LSTM-based versions. Both are trained on the KIT AIS vehicle dataset.

During inference the respective model is loaded in and initialized with the detected box information from the first frame. Then, during each frame the predicted locations of all objects are fetched using the motion model while all state properties for the current state are updated. The similarity is assessed again with the generalized IoU Loss which is then used for the matching with the Hungarian Method to associate to the tracklets.

Chapter 5

Experiments

5.1 Setup

The setup for all experimental Results listed in 5.2 is described here.
For all experiments, the configuration used in TransTrack is adopted to offer fair comparison to the baseline experiment whenever possible. The learning rate is set to 2e-4 and trained for 150 epochs with a batch size of 4. The coefficients for bounding box loss and generalized IoU loss are set to 5 and 2, respectively. Augmentations are used including random crop and horizontal flipping. The latter is adapted for providing flipped angles. The network uses weights pre-trained on the CrowdHuman dataset as it is done in the baseline.

5.2 Results

SEQ	IDF1 ↑	IDP ↑	IDR ↑	Rcll ↑	Prcn ↑	FAR ↓	GT	MT ↑	PT ↑	ML ↓	FP ↓	FN ↓	IDs ↓	FM ↓	MOTA ↑	MOTP ↑	MOTAL ↑	AE ↓
REGR	82.4	80.9	83.9	89.3	86.1	6.6	230	181	40	9	715	531	70	153	73.5	75.1	74.9	55.6
CLASS-CE	81.7	78.7	85	90	83.3	8.3	230	187	36	7	896	498	75	141	70.5	75.4	71.9	55.6
CLASS-FOC	81.5	79.9	83.2	88.3	84.7	7.3	230	179	40	11	791	584	66	166	71	75.1	72.3	54.6
ReID	86	85	87	90.9	88.8	5.3	230	199	20	11	570	453	48	87	78.5	82.1	79.4	-
LSTM	80.2	77.4	83.3	93.3	86.6	6.6	230	205	19	6	716	333	118	85	76.5	81.3	78.9	-

Table 5.1: Overview of results of all experiments.

To facilitate the description and interpretation of the visual results, the respective Figure is divided into four quadrants which will be denoted as *top-left*, *top-right*, *bottom-right* and *bottom-left*. The complete upper and lower halfs are referred to as *top* and *bottom*, respectively. The region in the center will be termed *middle*.
In Table 5.1 the summarized results of all experiments are listed. REGR, CLASS-CE and CLASS-FOC are the experiments for angle regression, regression and classification with Cross-Entropy Loss and with Focal Loss, respectively. ReID is the experiment with the re-Identification branch and LSTM is the motion model using the uni-directional LSTM. Since the angle experiments are conducted with OBBs they are not directly comparable to the others for which the usual HBBs are used.

5.2.1 Angle Error

This metric is used in addition to all metrics listed in Table 3.1 to evaluate experiments using OBBs. It calculates the deviation between predicted angle α_{pred} and ground-truth angle α_{gt}

while considering periodicity. With $d = | \alpha_{\text{pred}} - \alpha_{\text{gt}} |$ it is defined as follows:

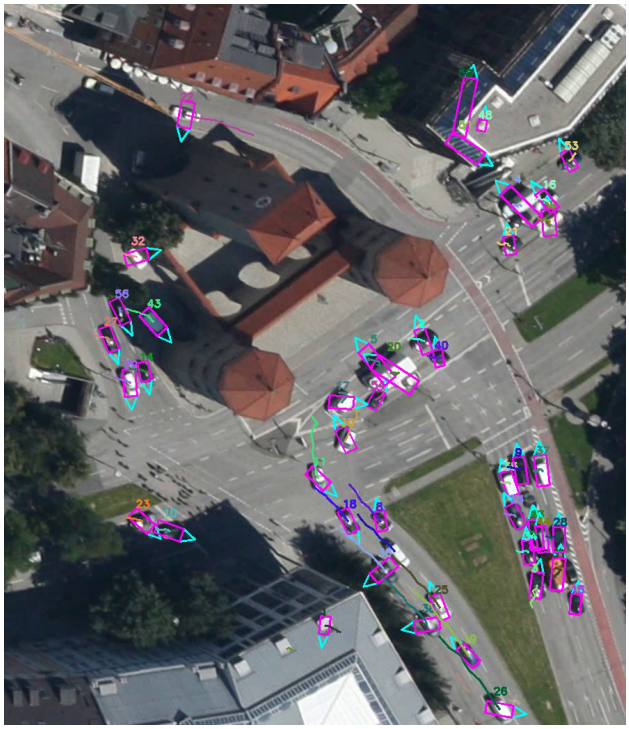
$$d = \begin{cases} d, & \text{if } x \leq 180 \\ 360 - d, & \text{otherwise} \end{cases} \tag{5.1}$$

5.2.2 Angle Regression

SEQ	IDF1 ↑	IDP ↑	IDR ↑	Rcll ↑	Prcn ↑	FAR ↓	GT	MT ↑	PT ↑	ML ↓	FP ↓	FN ↓	IDs ↓	FM ↓	MOTA ↑	MOTP ↑	MOTAL ↑	AE ↓
MC02	77.1	74.7	79.6	86.5	81.2	9.6	66	48	15	3	433	291	19	83	65.5	75.3	66.3	70.4
MS02	91.1	89.8	92.4	93.2	90.6	3.6	47	42	3	2	72	51	1	9	83.4	70.8	83.5	36.9
MS04	89.4	89.3	89.6	95.9	95.5	2.3	68	64	3	1	68	62	47	28	88.4	80.1	91.3	37.3
SC01	72.3	71.4	73.3	77.1	75	10.1	49	27	19	3	142	127	3	33	50.9	64.3	51.3	79.2
SUM	82.4	80.9	83.9	89.3	86.1	6.6	230	181	40	9	715	531	70	153	73.5	75.1	74.9	55.6

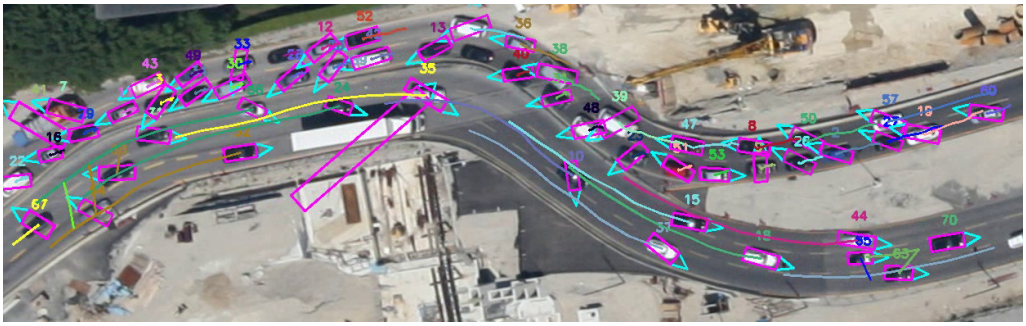
Table 5.2: Results for *Angle Regression with on OBB*

In this experiment the effect of the regression head is inspected. It contributes to the total loss and is assigned a coefficient of 1. The results are presented in Figure 5.1 and Table 5.2. Since the OBB ground-truth is different from the HBB one used in the TransTrack baseline experiment, the results are not directly comparable. The total MOTA score is 73.5 with a high performance on MS02 and MS04 while lacking points in the other two sequences. One noticeable property is the angle error which behaves very similarly to the MOTA score. The visual results explain this quite well. In Figure 5.1 b) and c) all boxes are oriented roughly in the correct direction. The trajectory path helps to see which direction the vehicle is going. The cars in the upper part of the image (e.g. 27, 30, 34, 16) are all facing the correctly to the left (ca. 180°) while the ones in the lower part are pointing correctly in the opposite (ca. 0°) direction. A similar scenario can be observed in c). In a) the boxes in the top-left and bottom-right are roughly facing in the correct direction while the top-right and center show many wrong cases. These cases are reflected in the higher FP and FN values. These in turn contribute in decreasing the MOTA score. It is notable that the IDSW behave differently. The switching occurs mostly in occasions where vehicle move close by each other, i.e. in MS04.



(a) Frame 10 of sequence MC02

(b) Frame 10 of sequence MS02



(c) Frame 10 of sequence MS04

Figure 5.1: Results for *Angle Regression on OBB*

5.2.3 Angle Regression & Classification

The classification head is placed next to the regression head and the robustness of the angle prediction is examined. The experiments are conducted with Cross-Entropy Loss and Focal Loss to classify the predicted angles to their respective sectors. Both losses are added to the total loss with a regression and classification coefficient of 1 and 0.5, respectively. The α and γ parameters in the Focal Loss are set to 0.25 and 2, respectively.

Cross-Entropy Loss

SEQ	IDF1 ↑	IDP ↑	IDR ↑	Rcll ↑	Prcn ↑	FAR ↓	GT	MT ↑	PT ↑	ML ↓	FP ↓	FN ↓	IDs ↓	FM ↓	MOTA ↑	MOTP ↑	MOTAL ↑	AE ↓
MC02	79.1	76.4	81.9	87.2	81.3	9.6	66	51	12	3	432	277	16	61	66.4	76.4	67	61.8
MS02	86.7	81.6	92.4	92.6	81.9	7.7	47	42	4	1	153	55	2	18	71.9	68.7	72.1	37.5
MS04	89	87.2	90.9	97.2	93.3	3.7	68	66	2	0	106	42	45	24	87.3	79.8	90.2	35.9
SC01	65.8	61.6	70.6	77.6	67.7	14.6	49	28	18	3	205	124	12	38	38.5	66.2	40.4	92.1
SUM	81.7	78.7	85	90	83.3	8.3	230	187	36	7	896	498	75	141	70.5	75.4	71.9	55.6

Table 5.3: Results for *Angle Regression and Classification with CE Loss on OBB*

A lower MOTA score of 70.5 can be seen in Table 5.3. The IDSW are not much higher than the regression experiment which means FP or FN counts must be higher. And indeed there are nearly 200 more FP (896 vs. 715) while the FN stays similar (498 vs. 531). In Figure 5.2 b) the three FP boxes on the roof are apparent. Even though the classification head is supposed to stabilize the orientation detection it leads to a worse detection performance (lower Prcn) while the total AE over all sequences stays the same. This can partially be explained by the network having to learn this additional task while not being able to learn the connection to the regression task. Over all sequences, the horizontally oriented vehicles (0° and 180°) seem to be detected best which can be seen especially in Figure b) and c). While the vehicles in the top-left of Figure a) driving towards the bottom, e.g. ID 17,18 and 20,35 are captured overall well the orientation of the cars on the bottom-right driving in the opposite direction (ID 14, 24) is not correctly detected at all. This could result from a low representation of these angles in the training dataset. In order to account for an imbalance in the dataset the experiment is conducted with Focal Loss.

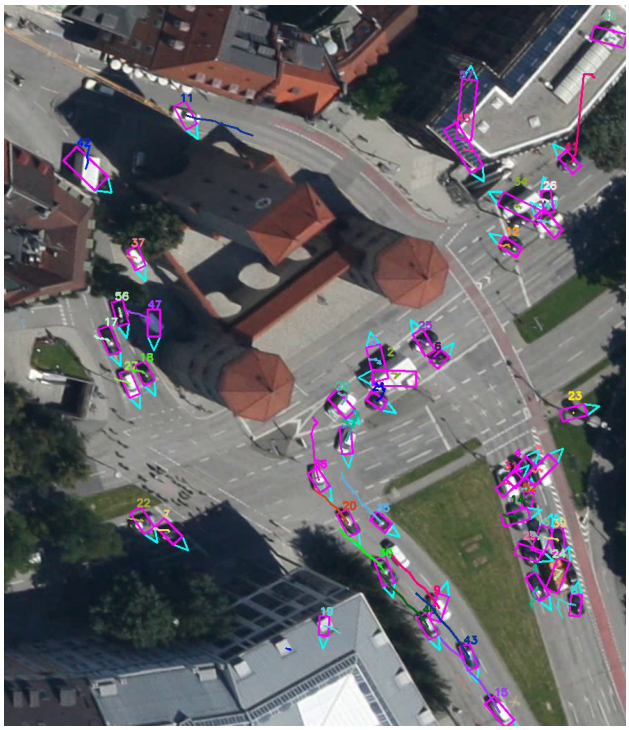
Focal Loss

SEQ	IDF1 ↑	IDP ↑	IDR ↑	Rcll ↑	Prcn ↑	FAR ↓	GT	MT ↑	PT ↑	ML ↓	FP ↓	FN ↓	IDs ↓	FM ↓	MOTA ↑	MOTP ↑	MOTAL ↑	AE ↓
MC02	78	76	80.2	86.3	81.7	9.2	66	49	15	2	416	295	19	79	66.1	75	67	67.4
MS02	90.3	88.7	91.8	92.9	89.8	4	47	42	4	1	79	53	2	17	82	69.1	82.2	42.9
MS04	87.8	87.4	88.2	93.9	93.1	3.7	68	62	5	1	106	92	37	35	84.5	80.5	86.9	39.5
SC01	66.6	64	69.3	74	68.3	13.6	49	26	16	7	190	144	8	35	38.3	66.6	39.5	69.7
SUM	81.5	79.9	83.2	88.3	84.7	7.3	230	179	40	11	791	584	66	166	71	75.1	72.3	54.6

Table 5.4: Results for *Angle Regression and Classification with Focal Loss on OBB*

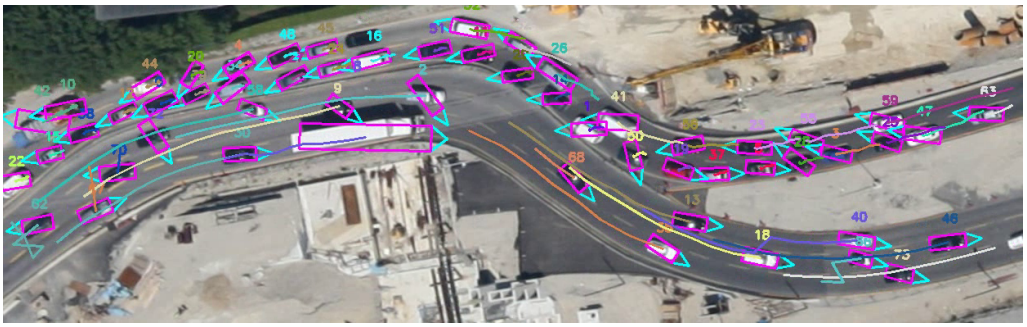
Table 5.4 shows the results using Focal Loss. The MOTA is 71 and thus slightly higher than the Cross-Entropy run but still lower than the regression case. The AE is more evenly distributed over the sequences with the lowest total AE in all runs. This indicates that the network tried to learn the difficult cases as well. This means the angles which are underrepresented in the training set. Even though the performance is far from acceptable this run provides the lowest AE in the most difficult sequence (SC01). Overall visually, the predicted boxes shown

in Figure 5.3 seem slightly worse than in the Cross-Entropy and Regression run. The AE numbers for these sequences confirm this. The main improvement is found in the SC01.



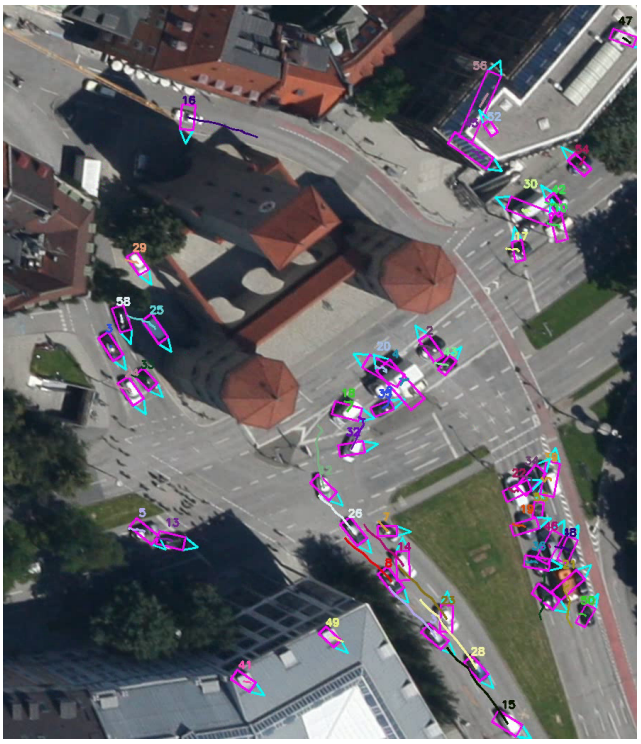
(a) Frame 10 of sequence MC02

(b) Frame 10 of sequence MS02



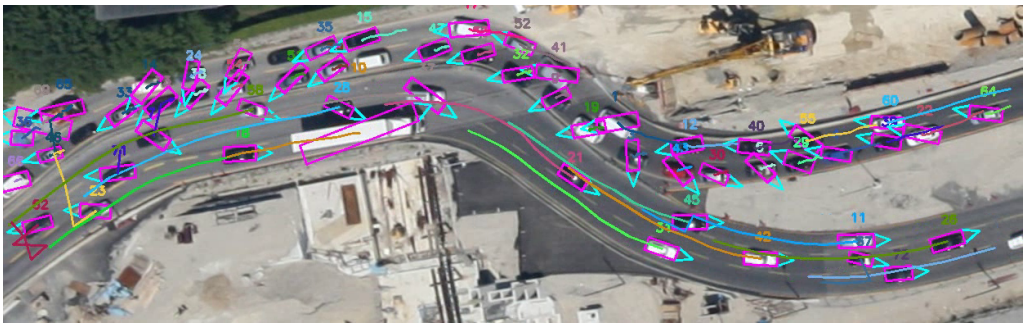
(c) Frame 10 of sequence MS04

Figure 5.2: Results for *Angle Regression and Classification with Cross-Entropy Loss on OBB*



(a) Frame 10 of sequence MC02

(b) Frame 10 of sequence MS02

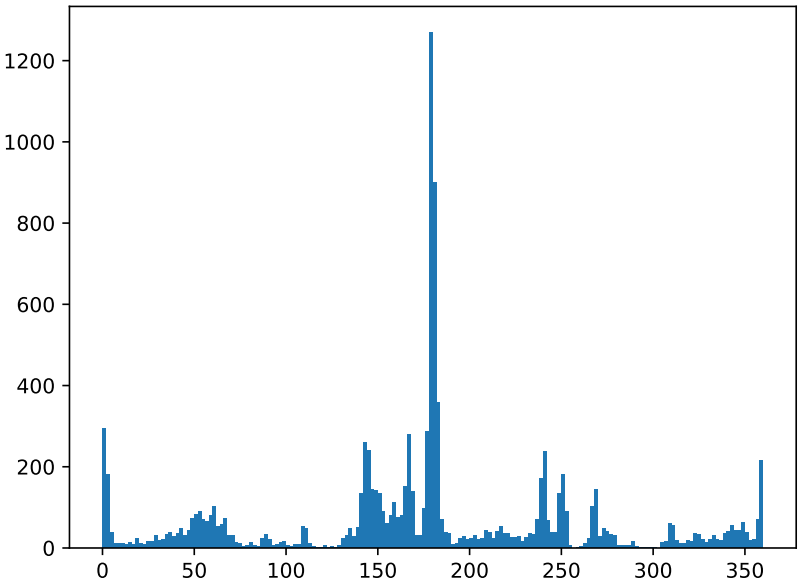


(c) Frame 10 of sequence MS04

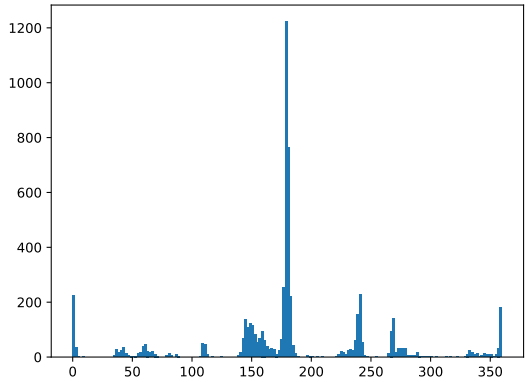
Figure 5.3: Results for *Angle Regression and Classification with Focal Loss on OBB*

5.2.4 Conclusion: Angle Experiments

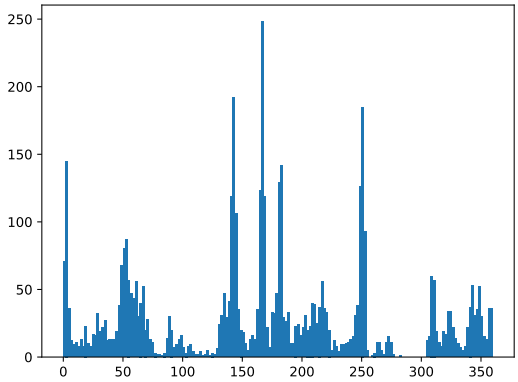
From these experiments it can be concluded that horizontally aligned vehicles (0° and 180°) are detected best. The network has difficulties predicting other angle ranges. One general problem contributing to the overall low prediction performance could be the quality of the sequences in the dataset. For HBB object detection, the resolution and recording quality are



(a) The complete dataset



(b) The training set



(c) The test set

Figure 5.4: Distribution of angles in KIT AIS

sufficient as the baseline shows. In order to properly predict the orientation visual features of the vehicles have to be rich. This means, e.g. to distinguish the front and back of a car. For the horizontally oriented instances this however is mostly not an issue in the results. Also, a high number of different vehicle instances are needed as well for a good generalization ability. For example, in the aerial object detection dataset DOTA v.1.5 [Xia+18] there exist 126k instances of small vehicles in the training set alone [Guo+20]. Compared to that the KIT AIS training set only holds 5842 annotated vehicles. Although this number is not remotely comparable to DOTA, the number should suffice for substantially better prediction than presented in the results. The main problem has to lie somewhere else.

To investigate this issue, the *distribution of angles* in the KIT AIS dataset is inspected. In Figure 5.4 a) the distribution of angles in the whole dataset can be seen. What immediately stands out is the high frequency around the 180° angle. The next most frequent orientation lies at

0°/360° and some peaks can be seen at around 140° and 240°. An optimal dataset should have all angles uniformly distributed. This is of course difficult to achieve with non-synthetic data. Still, this extreme imbalance does not provide a good base for the experiments. In order to draw conclusions on the network performance both training and test set angle distributions have to be considered which are presented in Figure 5.4 b) and c), respectively. The distribution in b) is overall very similarly imbalanced as in the complete dataset. This explains the decent prediction of the 0° and 180° angles as the models have learned their relation with the vehicle visual features. The rest of the angles are likely to be seen as outliers. In c) however, the distribution is much more even with 4-5 frequent angle ranges. This makes the prediction extremely hard. There are approaches that may help improve the prediction results [Yi+21] [Han+21]. However, considering the data does not provide a sufficient baseline for training these potential improvements are probably marginally and do not solve the root issue.

The initial motivation for the angle prediction was the improvement of the tracking capability. A realistic angle deviation (<10°) of a vehicle in two consecutive frames was supposed to serve as a constraint in both training and inference. For that, a precise angle prediction was the condition which these experiments did not show. Thus, these methods are no longer pursued.

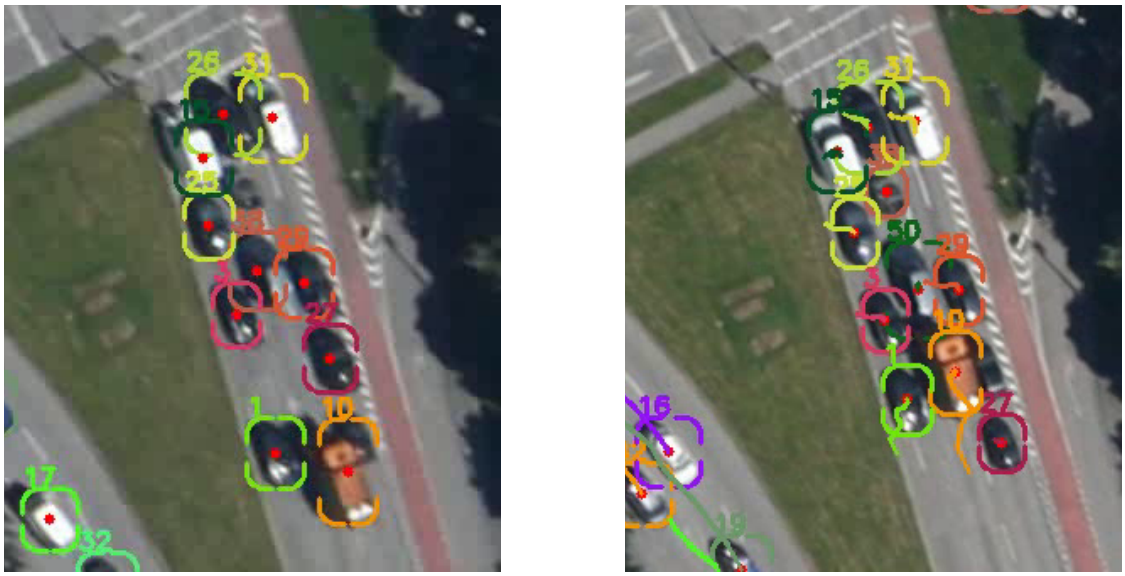
5.2.5 Re-ID Branch

SEQ	IDF1 ↑	IDP ↑	IDR ↑	Rcll ↑	Prcn ↑	FAR ↓	GT	MT ↑	PT ↑	ML ↓	FP ↓	FN ↓	IDs ↓	FM↓	MOTA ↑	MOTP ↑	MOTAL ↑
MC02	79.3	78.8	79.8	85.9	84.8	7.4	66	51	10	5	331	305	14	43	69.8	82.3	70.4
MS02	91.8	92.6	91	92.4	94	2.2	47	41	4	2	44	57	7	21	85.5	82.6	86.3
MS04	94.2	91.6	97	98.4	92.8	4	68	68	0	0	116	25	12	17	89.9	83.6	90.6
SC01	81	80.1	82	88.1	86.1	5.6	49	39	6	4	79	66	15	6	71.1	76.3	73.6
SUM	86	85	87	90.9	88.8	5.3	230	199	20	11	570	453	48	87	78.5	82.1	79.4

Table 5.5: Results for *Re-ID branch with Circle Loss on HBB*

The experiments for the re-ID branch are conducted with a circle loss coefficient of 0.004. The results are shown in Table 5.5 and Figure 5.6. These can be directly compared to the TransTrack baseline presented in Table 3.5 and Figure 3.2, respectively as they were trained with the same HBBs. The truck (ID 41, red) in the top-left of Figure 5.6 a) exhibits a trajectory indicating that it has been tracked for a few frames. The baseline counterpart of the vehicle (ID 59, beige) is detected but does not show the trajectory line which means the baseline model was not able to consistently track it for the last frames. On the roofs less FP can be seen in this experiment. However, some cars are not detected, e.g. in the top-left. Viewing Table 5.5, these visual findings can be supported by the increased MOTA from 78.1 to 78.5. The FP are lower (570 vs. 716) while the FN are higher (453 vs. 333). This in turn means that Prcn is higher (88.8 vs. 86.6).

This experiment was motivated by the IDSW occurring with the baseline model which was described in Subsection 4.2.1. There, the identity of the (orange) vehicle had switched. Naturally, it is interesting to see whether this happens in the re-ID variant as well. For that, Figure 5.5 is considered which shows the same (cropped) frames as in Figure 4.1 but with the visualization of this experiment. Fortunately, the orange vehicle has the ID 10 both in the

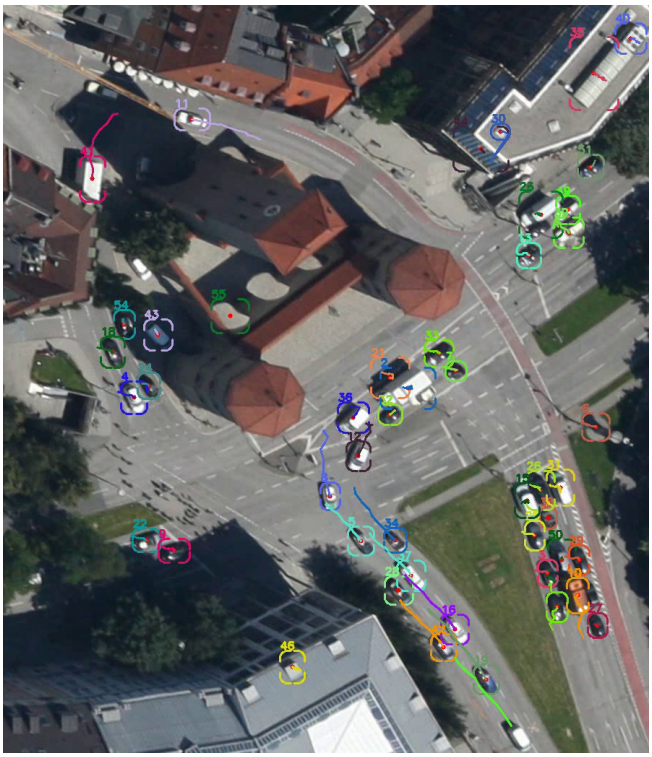


(a) Frame 1 (crop) **(b)** Frame 10 (crop)

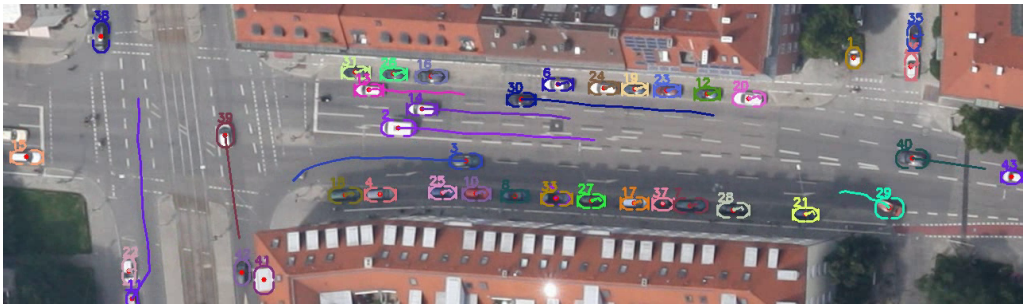
Figure 5.5: Inspection of IDSW: Results with *Re-ID branch* on MC02

first frame and also in the 10th frame. The re-ID branch was therefore able to prevent this IDSW from happening. What stands out is that in sequence MS04, all 68 identities are mostly tracked with half as many IDSW compared to the baseline (12 vs. 24). However, over all sequences the total number of IDSW slightly increased. MS04 does not display a scenario which offers many possibilities for false positives. In contrast to that, MC02 and MS02 both show many rectangular-shaped blocks on the rooftops which are very challenging to the detector. The Transformer detection branch has a certain amount of queries (here 500) that act as potential object candidates. The more demanding the visual scenario is, the higher the FP and FN. The high FP and FN values in MC02 and MS02 also explain the occurrence of more IDSW. Rooftops that look similar to vehicles provide the possibility of the re-identification mistaking it and creating an IDSW. This shows, the re-ID is influenced by the detection performance.

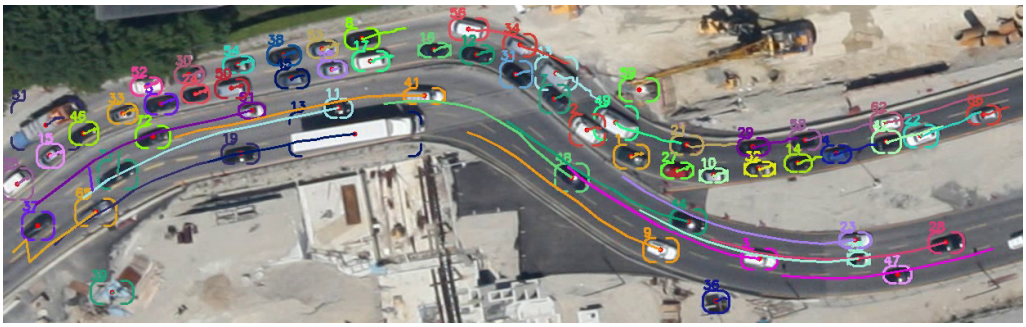
To further explain these results, the modification to the baseline architecture is considered. The Re-ID version calculates the similarity of all pairs of detection and track box features. If a detection and track box pair is feature-wise very similar then it results in a match. This helps even if the track box is not located precisely. The trajectory can therefore be continued with the correct ID and less IDSW are the result. On the other hand, the predicted location by the tracking head may be very accurate but if its features are not similar it would match with a detection that is more distant but feature-wise more alike. This leads to less IDs. The reason for a sub-optimal feature representation could lie in the somewhat low visual quality of the dataset.



(a) Frame 10 of sequence MC02



(b) Frame 10 of sequence MS02



(c) Frame 10 of sequence MS04

Figure 5.6: Results for *Re-ID Branch with Circle Loss on HBB*

5.2.6 LSTM

The LSTM motion model is trained for 400 epochs on the KIT AIS dataset. The results can be found in Table 5.6 and Figure 5.7. The Bi-directional LSTM achieved nearly the same results which is why it is omitted at this point.

SEQ	IDF1 ↑	IDP ↑	IDR ↑	Rcll ↑	Prcn ↑	FAR ↓	GT	MT ↑	PT ↑	ML ↓	FP ↓	FN ↓	IDs ↓	FM↓	MOTA ↑	MOTP ↑	MOTAL ↑
MC02	73.2	69.1	77.8	90.8	80.6	10.4	66	56	8	2	470	198	48	44	66.8	82.3	68.9
MS02	89.6	88.8	90.5	94.2	92.5	2.9	47	44	2	1	57	43	9	11	85.4	79.8	86.5
MS04	85.4	83.6	87.3	97.1	92.9	3.9	68	63	5	0	112	44	48	23	86.6	81.8	89.6
SC01	82.2	80.1	84.3	91.3	86.8	5.5	49	42	4	3	77	48	13	7	75.1	78.4	77.2
SUM	80.2	77.4	83.3	93.3	86.6	6.6	230	205	19	6	716	333	118	85	76.5	81.3	78.9

Table 5.6: Results for *LSTM Motion Prediction on HBB*

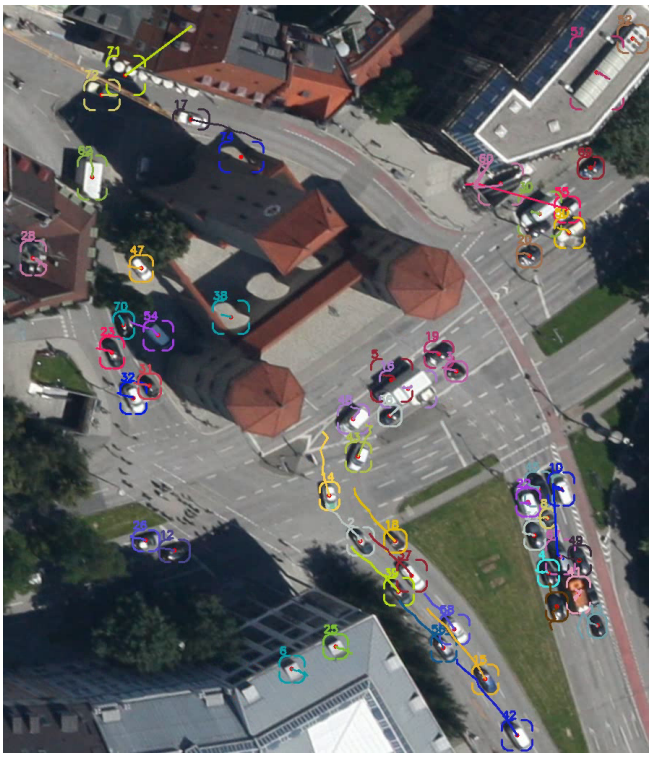
Overall, the visualization in Figure 5.7 seems similar to the baseline. However, one thing to notice are some occurrences of sharp "jumps" (ID Transfer) of some vehicles recognizable by straight lines. This can be observed in e.g. the top-left and top-right of a) by yellow and red lines, respectively. In c) this is apparent by the many "zig-zag" lines. This indicates a high number of IDs which the Table 5.6 confirms. There are 118 IDs compared to 41 in the baseline. The Bi-directional version exhibits 120 IDs. Considering the fact that the motion model is supposed to predict a trajectory's next position this might seem strange. This however can be explained by the track rebirth feature. Unmatched track boxes are saved and can be "rebirthed" when the IoU similarity to a detection is high enough. Without this property the tracking performance suffers significantly. Another reason is simply the lower prediction performance of the LSTM.

Since only the motion prediction was replaced in this experiment, the effect of the original tracking branch can be well identified. With a MOTA of 76.5 the score is only lower by 1.6 compared to the baseline.

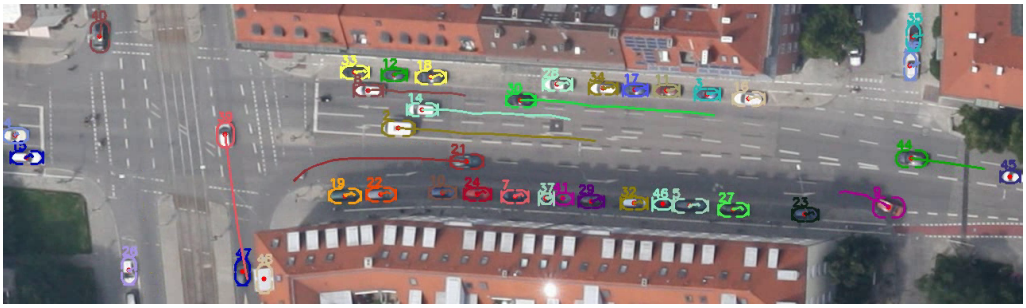
5.2.7 Conclusion: Re-ID and LSTM

The Re-ID experiment shows an improved MOTA score of 78.5 compared to the baseline (78.1). Even though two sequences show lower IDSW, the total number slightly increased to 48. The re-identification works but also hinders finding a match between detection and track box when the feature representation is poor. This might stem from the rather low visual quality of the dataset. Also, IDSW can occur with an increase of FP an FN. In fact, it could be also the overall improved detection performance resulting in the higher MOTA score. The detection head might benefit slightly from the re-ID loss and from having a shared Transformer encoder.

The LSTM experiment resulted in a lower MOTA of 76.5. The location prediction is not precise enough to exceed the baseline performance. This shows the efficacy of the the performance of the tracking branch which simultaneously learns about location and visual features to predict the object in the current frame. On the other hand, the MOTA being lower by only 1.6 points expresses the strong performance of the common components: The detection branch together with the rest of the TransTrack architecture. This shows the Deformable DETR contributes a lot to the tracking performance.



(a) Frame 10 of sequence MC02



(b) Frame 10 of sequence MS02

(c) Frame 10 of sequence MS04

Figure 5.7: Results for *LSTM*

Chapter 6

Conclusion

This thesis tackled the task of multi-vehicle tracking in top-down aerial imagery sequences. A benchmark on a selection of algorithms was made displaying their performance on the KIT AIS dataset. It was shown that MOT methods designed for specific domains do not automatically translate to others. Some methods failed completely to provide basic vehicle tracking capability. DEFT, CenterTrack and TransTrack were able to tackle this task with varying performance. TransTrack, with its Transformer-based architecture delivered the best results for aerial MOT. Using it as the basis, a variety of modifications were made to inspect their effect on multi-vehicle tracking. The prediction of the vehicle orientation appears to work approximately only in simple cases, i.e horizontal orientation. The angle error is the lowest when combining regression and classification using the Focal Loss. However, the highly imbalanced dataset w.r.t the angle distribution prohibits the network from learning a precise prediction. The vehicle re-identification branch achieves the highest MOTA score with 78.5. This feature was motivated by the occurrence of identity switches in the baseline. The re-ID branch is able to reduce identity switches in one sequence drastically. This sequence does not contain many visual objects that could be mistaken with vehicles, e.g. rectangular-shaped rooftops. This leads to both lower FP and FN. These in turn reduce the possibility of false re-identifications. With more challenging sequences the IDSW rises. The re-ID capability thus also depends a lot on the detector. The higher MOTA score might also benefit from a overall stronger detection capability. The motion model shows the superiority of the tracking branch in the base algorithm. Since the LSTM results in a MOTA score of only 1.6 points lower than the baseline, the majority of the tracking capability can also be attributed to the independent detection branch. In fact, the architecture excluding the tracking branch can be viewed as the Deformable DETR which in the end may be responsible for largely the high MOTA score.

The findings of this thesis can be utilized for further work in Multi-Object Tracking applied in top-down aerial imagery sequences. To address the angle prediction problem, the Transformer-based algorithm is still suitable as a recent DETR-based work tackled oriented object detection [Ma+21]. However, it has to be trained on a larger dataset first with a more regular angle distribution to achieve more precise angle predictions. Due to the lack of top-down aerial MOT datasets, an object detection dataset would have to be used. For that, the existing joint-tracking-and-detection architecture has to be reduced to an object detection network for pre-training. Object re-identification networks may help to further improve the prevention of identity switches. Completely different architectures could be integrated coupled with other loss functions. To improve tracking features, a greater number of past frames could be consulted to form more temporally stable visual features.

List of Figures

List of Tables

Bibliography

[Azi+21] Azimi, S. M., Bahmanyar, R., Henry, C., and Kurz, F. "EAGLE: Large-Scale Vehicle Detection Dataset in Real-World Scenarios using Aerial Imagery". In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 6920–6927.

[Azi+20] Azimi, S. M., Kraus, M., Bahmanyar, R., and Reinartz, P. "Multiple Pedestrians and Vehicles Tracking in Aerial Imagery: A Comprehensive Study". In: *ArXiv* abs/2010.09689 (2020).

[BAR19] Bahmanyar, R., Azimi, S., and Reinartz, P. "Multiple vehicle and people tracking in aerial imagery using stack of micro single-object-tracking CNNs". In: ISPRS. 2019.

[BS08] Bernardin, K. and Stiefelhagen, R. "Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics". In: *J. Image Video Process.* 2008 (Jan. 2008). ISSN: 1687-5176. DOI: 10.1155/2008/246309. URL: https://doi.org/10.1155/2008/246309.

[Bis06] Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.

[BES17] Bochinski, E., Eiselein, V., and Sikora, T. "High-speed tracking-by-detection without using image information". In: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE. 2017, pp. 1–6.

[CV18] Cai, Z. and Vasconcelos, N. "Cascade r-cnn: Delving into high quality object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 6154–6162.

[Car+20] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. "End-to-End Object Detection with Transformers". In: *Computer Vision – ECCV 2020*. Ed. by Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M. Cham: Springer International Publishing, 2020, pp. 213–229. ISBN: 978-3-030-58452-8.

[Cha+21] Chaabane, M., Zhang, P., Beveridge, J. R., and O'Hara, S. "DEFT: Detection Embeddings for Tracking". In: *CoRR* abs/2102.02267 (2021). arXiv: 2102.02267. URL: https://arxiv.org/abs/2102.02267.

[Che+20] Chen, J., Wu, Q., Liu, D., and Xu, T. "Foreground-Background Imbalance Problem in Deep Object Detectors: A Review". In: *2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)* (2020), pp. 285–290.

[Che+18] Chen, L., Ai, H., Zhuang, Z., and Shang, C. "Real-Time Multiple People Tracking with Deeply Learned Candidate Selection and Person Re-Identification". In: *2018 IEEE International Conference on Multimedia and Expo (ICME)* (July 2018). DOI: 10.1109/icme.2018.8486597. URL: http://dx.doi.org/10.1109/ICME.2018.8486597.

[Dai+17] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., and Wei, Y. “Deformable convolutional networks”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 764–773.

[Don+20] Dong, X., Wang, P., Zhang, P., and Liu, L. “Probabilistic oriented object detection in automotive radar”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2020, pp. 102–103.

[Dua+19] Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., and Tian, Q. “CenterNet: Keypoint Triplets for Object Detection”. In: *CoRR* abs/1904.08189 (2019). arXiv: 1904.08189. URL: http://arxiv.org/abs/1904.08189.

[Fan+20] Fan, H., Du, D., Wen, L., Zhu, P., Hu, Q., Ling, H., Shah, M., Pan, J., Schumann, A., Dong, B., et al. “VisDrone-MOT2020: The Vision Meets Drone Multiple Object Tracking Challenge Results”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 713–727.

[GLU12] Geiger, A., Lenz, P., and Urtasun, R. “Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[Gir15] Girshick, R. “Fast r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448.

[Gir+14] Girshick, R., Donahue, J., Darrell, T., and Malik, J. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[Guo+20] Guo, W., Li, W., Gong, W., and Cui, J. “Extended Feature Pyramid Network with Adaptive Scale Training Strategy and Anchors for Object Detection in Aerial Images”. In: *Remote Sensing* 12 (Mar. 2020), p. 784. DOI: 10.3390/rs12050784.

[Han+21] Han, J., Ding, J., Xue, N., and Xia, G.-S. “Redet: A rotation-equivariant detector for aerial object detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 2786–2795.

[HNS19] Hancock, P. A., Nourbakhsh, I., and Stewart, J. “On the future of transportation in an era of automated and autonomous vehicles”. In: *Proceedings of the National Academy of Sciences* 116.16 (2019), pp. 7684–7691.

[HVC17] Hara, K., Vemulapalli, R., and Chellappa, R. “Designing deep convolutional neural networks for continuous object orientation estimation”. In: *arXiv preprint arXiv:1702.01499* (2017).

[HTS16] Held, D., Thrun, S., and Savarese, S. “Learning to track at 100 fps with deep regression networks”. In: *European conference on computer vision*. Springer. 2016, pp. 749–765.

[HBL17] Hermans, A., Beyer, L., and Leibe, B. “In defense of the triplet loss for person re-identification”. In: *arXiv preprint arXiv:1703.07737* (2017).

[HS97] Hochreiter, S. and Schmidhuber, J. “Long Short-Term Memory”. In: 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[KVB88] Kanopoulos, N., Vasanthavada, N., and Baker, R. L. “Design of an image edge detection filter using the Sobel operator”. In: *IEEE Journal of solid-state circuits* 23.2 (1988), pp. 358–367.

[21a] *KIT AIS Data Set*. Aug. 31, 2021. URL: https://www.ipf.kit.edu/downloads_data_set_AIS_vehicle_tracking.php.

[KSH12] Krizhevsky, A., Sutskever, I., and Hinton, G. E. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012), pp. 1097–1105.

[Li+18] Li, B., Yan, J., Wu, W., Zhu, Z., and Hu, X. "High performance visual tracking with siamese region proposal network". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8971–8980.

[Lia+21] Liang, C., Zhang, Z., Zhou, X., Li, B., Lu, Y., and Hu, W. "One More Check: Making "Fake Background" Be Tracked Again". In: *CoRR* abs/2104.09441 (2021). arXiv: 2104.09441. URL: https://arxiv.org/abs/2104.09441.

[Lin+17] Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. "Focal Loss for Dense Object Detection". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2999–3007. DOI: 10.1109/ICCV.2017.324.

[Liu+16] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. "Ssd: Single shot multibox detector". In: *European conference on computer vision*. Springer. 2016, pp. 21–37.

[Lui+20] Luiten, J. T., Osep, A., Dendorfer, P., Torr, P., Geiger, A., Leal-Taixé, L., and Leibe, B. "HOTA: A Higher Order Metric for Evaluating Multi-object Tracking". In: *International journal of computer vision* (2020). Published: 08 October 2020. ISSN: 1573-1405. DOI: 10.1007/s11263-020-01375-2. URL: https://publications.rwth-aachen.de/record/804671.

[Luo+20] Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., and Kim, T.-K. "Multiple object tracking: A literature review". In: *Artificial Intelligence* (2020), p. 103448.

[Ma+21] Ma, T., Mao, M., Zheng, H., Gao, P., Wang, X., Han, S., Ding, E., Zhang, B., and Doermann, D. "Oriented Object Detection with Transformer". In: *arXiv preprint arXiv:2106.03146* (2021).

[Mil+16a] Milan, A., Leal-Taixé, L., Reid, I., Roth, S., and Schindler, K. "MOT16: A Benchmark for Multi-Object Tracking". In: *arXiv:1603.00831 [cs]* (Mar. 2016). arXiv: 1603.00831. URL: http://arxiv.org/abs/1603.00831.

[Mil+16b] Milan, A., Leal-Taixé, L., Reid, I. D., Roth, S., and Schindler, K. "MOT16: A Benchmark for Multi-Object Tracking". In: *ArXiv* abs/1603.00831 (2016).

[Mil+16c] Milan, A., Rezatofighi, S. H., Dick, A. R., Schindler, K., and Reid, I. D. "Online Multi-target Tracking using Recurrent Neural Networks". In: *CoRR* abs/1604.03635 (2016). arXiv: 1604.03635. URL: http://arxiv.org/abs/1604.03635.

[21b] *MOT2016/2017 Evaluation Tool*. Nov. 7, 2021. URL: https://github.com/shenh10/mot_evaluation.

[Nie15] Nielsen, M. A. *Neural networks and deep learning*. Vol. 25. Determination press San Francisco, CA, 2015. Chap. 3.

[RCC17] Ranjan, R., Castillo, C. D., and Chellappa, R. "L2-constrained softmax loss for discriminative face verification". In: *arXiv preprint arXiv:1703.09507* (2017).

[Red+16] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[Ren+15] Ren, S., He, K., Girshick, R., and Sun, J. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015), pp. 91–99.

[Ris+16] Ristani, E., Solera, F., Zou, R. S., Cucchiara, R., and Tomasi, C. "Performance Measures and a Data Set for Multi-target, Multi-camera Tracking". In: *ArXiv* abs/1609.01775 (2016).

[Sha+18] Shao, S., Zhao, Z., Li, B., Xiao, T., Yu, G., Zhang, X., and Sun, J. "CrowdHuman: A Benchmark for Detecting Human in a Crowd". In: *arXiv preprint arXiv:1805.00123* (2018).

[21c] *Situation information for disaster management*. Nov. 13, 2021. URL: https://www.dlr.de/content/en/articles/news/2021/03/20210716_situation-information-for-disaster-management.html.

[SSB20] Stadler, D., Sommer, L. W., and Beyerer, J. "PAS tracker: position-, appearance- and size-aware multi-object tracking in drone videos". In: *European Conference on Computer Vision*. Springer. 2020, pp. 604–620.

[Sun+21] Sun, P., Cao, J., Jiang, Y., Zhang, R., Xie, E., Yuan, Z., Wang, C., and Luo, P. *TransTrack: Multiple Object Tracking with Transformer*. 2021. arXiv: 2012.15460 [cs.CV].

[SWT15] Sun, Y., Wang, X., and Tang, X. "Deeply learned face representations are sparse, selective, and robust". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 2892–2900.

[Sun+20] Sun, Y., Cheng, C., Zhang, Y., Zhang, C., Zheng, L., Wang, Z., and Wei, Y. "Circle loss: A unified perspective of pair similarity optimization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 6398–6407.

[21d] *TORCH.ATAN2*. Oct. 30, 2021. URL: https://pytorch.org/docs/stable/generated/torch.atan2.html#torch.atan2.

[Vas+17] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[Wan+18] Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., and Liu, W. "Cosface: Large margin cosine loss for deep face recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 5265–5274.

[Wan+17] Wang, J., Zhou, F., Wen, S., Liu, X., and Lin, Y. "Deep metric learning with angular loss". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2593–2601.

[Wan+19] Wang, Q., Zhang, L., Bertinetto, L., Hu, W., and Torr, P. H. "Fast online object tracking and segmentation: A unifying approach". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1328–1338.

[Wan+20] Wang, Z., Zheng, L., Liu, Y., Li, Y., and Wang, S. "Towards real-time multi-object tracking". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*. Springer. 2020, pp. 107–122.

[WB06] Welch, G. and Bishop, G. "An Introduction to the Kalman Filter". In: *Proc. Siggraph Course* 8 (Jan. 2006).

[WBP17] Wojke, N., Bewley, A., and Paulus, D. "Simple online and realtime tracking with a deep association metric". In: *2017 IEEE international conference on image processing (ICIP)*. IEEE. 2017, pp. 3645–3649.

[Wol+20] Wolf, T., Chaumond, J., Debut, L., Sanh, V., Delangue, C., Moi, A., Cistac, P., Funtowicz, M., Davison, J., Shleifer, S., et al. "Transformers: State-of-the-art natural language processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 2020, pp. 38–45.

[Wu+21] Wu, J., Cao, J., Song, L., Wang, Y., Yang, M., and Yuan, J. "Track to Detect and Segment: An Online Multi-Object Tracker". In: *CoRR* abs/2103.08808 (2021). arXiv: 2103.08808. URL: https://arxiv.org/abs/2103.08808.

[Xia+18] Xia, G.-S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M., and Zhang, L. "DOTA: A large-scale dataset for object detection in aerial images". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 3974–3983.

[Yan+18] Yang, X., Sun, H., Fu, K., Yang, J., Sun, X., Yan, M., and Guo, Z. "Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation dense feature pyramid networks". In: *Remote Sensing* 10.1 (2018), p. 132.

[YYH20] Yang, X., Yan, J., and He, T. "On the Arbitrary-Oriented Object Detection: Classification based Approaches Revisited". In: *arXiv preprint arXiv:2003.05597* (2020).

[Yi+21] Yi, J., Wu, P., Liu, B., Huang, Q., Qu, H., and Metaxas, D. "Oriented object detection in aerial images with box boundary-aware vectors". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021, pp. 2150–2159.

[Zha+20] Zhang, Y., Wang, C., Wang, X., Zeng, W., and Liu, W. "A Simple Baseline for Multi-Object Tracking". In: *CoRR* abs/2004.01888 (2020). arXiv: 2004.01888. URL: https://arxiv.org/abs/2004.01888.

[Zho+19] Zhou, K., Yang, Y., Cavallaro, A., and Xiang, T. "Omni-scale feature learning for person re-identification". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 3702–3712.

[ZKK20] Zhou, X., Koltun, V., and Krähenbühl, P. "Tracking objects as points". In: *European Conference on Computer Vision*. Springer. 2020, pp. 474–490.

[Zhu+18] Zhu, P., Wen, L., Bian, X., Ling, H., and Hu, Q. "Vision meets drones: A challenge". In: *arXiv preprint arXiv:1804.07437* (2018).

[Zhu+20] Zhu, P., Wen, L., Du, D., Bian, X., Hu, Q., and Ling, H. "Vision Meets Drones: Past, Present and Future". In: *CoRR* abs/2001.06303 (2020). arXiv: 2001.06303. URL: https://arxiv.org/abs/2001.06303.

[Zhu+21] Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. *Deformable DETR: Deformable Transformers for End-to-End Object Detection*. 2021. arXiv: 2010.04159 [cs.CV].

[ZDW20] Zhu, Y., Du, J., and Wu, X. "Adaptive period embedding for representing oriented objects in aerial images". In: *IEEE Transactions on Geoscience and Remote Sensing* 58.10 (2020), pp. 7247–7257.