

Machine learning based multi-object tracking without dynamic models and hard association metrics

Christian Alexander Holz, Christian Bader, Matthias Drüppel

Abstract—In this paper, we develop Machine Learning (ML)-based methods for Multi Object Tracking (MOT) within the context of Advanced Driver Assistance Systems (ADAS). Given the increasing complexity and demand for precise and efficient object tracking systems in the automotive industry, this work focuses on the integration of ML techniques into established tracking methodologies. Key contributions encompass the creation and evaluation of three specialized neural networks: (i) the Single Prediction Network (SPENT) for predicting the trajectories of tracked objects, (ii) the Single Association Network (SANT) for associating incoming sensor objects with existing tracks, (iii) and the Multi Association Network (MANTA) for associating multiple sensor objects with existing tracks. Figure ?? provides an overview of our approach (i) and (ii). These networks aim to combine ML methods with a traditional Kalman filter framework, offering a data driven approach to addressing MOT challenges. We integrate our three ML networks into a Kalman framework and evaluate the performance, both of the components itself and the overall system. By replacing single components, we get a clearer understanding of the impact of the ML models on the overall tracking system. This approach also leaves the modularity of system intact while enabling machine learning (ML) for certain tracking tasks. The results reveal a modular, robust, and maintainable tracker, underscoring the potential of ML integration in ADAS.

Index Terms—Article submission, IEEE, IEEEtran, journal, LATEX, paper, template, typesetting.

I. INTRODUCTION

THE ongoing evolution of Advanced Driver Assistance Systems (ADAS) has brought the need for precise and reliable Multi Object Tracking (MOT) into the spotlight [1] [2] [3] [4] [5] [6] [7] [8]. In complex and dynamic environments, as encountered in urban traffic, it is crucial to simultaneously and accurately capture the positions and movements of multiple objects. Tracking multiple objects is a key challenge in computer vision (CV).

In the commonly used Tracking-by-Detection (TbD) paradigm, a tracker fuses detected sensor objects (SO) to create consistent object tracks over time. A key challenge within this paradigm is associating incoming measurements SO with their corresponding existing object tracks or initializing new object tracks.

Offline methods [9] process the entire video material at once in a batch process. However, these methods are unsuitable for most real-time applications, such as ADAS. In such applications, it is crucial to predict the state of objects immediately after new detections. Therefore, most recent approaches for tracking multiple objects rely on online methods that do not depend on future image information. Online methods use various features to estimate the similarity between the recognised objects and the existing tracks. This can be done on the basis of predicted positions or similarities in appearance [2] [3] [4] [5]. While some approaches only consider the most recent recognition corresponding to a track, other methods integrate temporal information into a track history. For example, methods use recurrent neural networks (RNNs) [4] [5] or attention mechanisms [6] [7] to aggregate temporal information.

For state prediction, in many TbD approaches, Kalman filters and their variants have proven to be effective. However, they reach their limits in more complex scenarios, particularly in the presence of non-linear motion patterns and interactions among multiple objects. In this work, we introduce a novel MOT approach that leverages Machine Learning (ML) to overcome these challenges. We specifically focus on the development and implementation of Neural Networks (NN), which can enable more precise and flexible data-driven object tracking without relying on cumbersome heuristics and hyperparameters.

The data association is typically carried out based on similarity scores calculated between the measurements and the existing tracks. These similarity scores may rely on the latest detection or be aggregated from historical detections. As in Mertz et al [5], the aim of this work was to develop a data-based approach that can learn to completely solve the combinatorial Non Deterministic Polynomial Time (NP) hard optimisation problem of data association. Mertz et al [5] use a distance matrix based on the Euclidean distance measure as input data for the developed DA network, thus replacing an association algorithm such as the Hungarian Algorithm (HA). It can be assumed that the Euclidean distance measure was used as the basis for calculating the ground truth (GT) training data (distance matrices) and for the evaluation. However, this is not explicitly stated. It can therefore be argued that this calculation step deprives the network of the opportunity to follow a different association logic or to learn it on the basis of data. In the context of this work, the hypothesis was therefore put forward that a Gated Recurrent Unit (GRU)-

C. Holz is with Daimler Truck AG, Research and Advanced Development, Stuttgart, Germany

C. Bader is with Daimler Truck AG, Research and Advanced Development, Stuttgart, Germany

M. Drüppel is with the Center for Artificial Intelligence, Duale Hochschule Baden-Württemberg (DHBW), Stuttgart, Germany

based association network can be formed by an undefined distance measure to increase the assignment of the temporal memory component and thus of the history.

The aim of this work was therefore to develop an association network that is intended to solve the assignment of one or more sensor objects (SO) to an existing number of object tracks without a defined distance measure.

II. RELATED WORK

Our primary contribution is the development and evaluation of three NN that we labeled: (i) the Single Prediction Network (SPENT), (ii) the Single Association Network (SANT), and (iii) the Multi Association Network (MANTA). Figure 1 provides an overview of our approach (i) and (ii). The proposed Single Prediction Network (SPENT) processes the sensor objects (SO) per time step, which consists of a state vector and contains information such as object position, orientation and dimension. SPENT predicts a fixed-dimensional state vector for each SO based on the received state vectors per SO. The output predictions from SPENT are used as input to the Single Association Network (SANT) to build target trajectories or object tracks. In the remainder of the following sections, we will explain the proposed modules in detail.

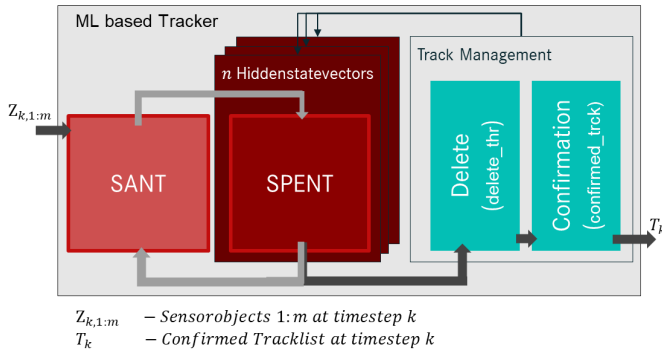


Fig. 1. Schematic representation of the two integrated networks SPENT and SANT in a tracking-by-detection (TbD) framework.

In comparison to the Kalman filter, SPENT is capable of predicting the state of individual objects without the need for a predefined state or observation model at runtime. SPENT holds the potential, particularly in terms of adaptability to various scenarios and the ability to effectively handle nonlinearities. Many conventional tracking systems rely on static methods for data association, often based on simple heuristics or fixed thresholds. In contrast, SANT employs machine learning to automate these processes and adapt more effectively to different scenarios. As a result, within the TbD MOT framework, SANT replaces the calculation of a distance metric and the Hungarian algorithm for the corresponding assignment. Furthermore, we integrate SPENT and SANT into an existing tracking system and demonstrate their performance through multiple tests and comparisons with established methods. This work provides exciting insights and a significant advancement in the development of Advanced Driver Assistance Systems (ADAS), contributing to the further evolution of technologies for autonomous driving (AD).

III. TRACKING WITH PREDICTION AND ASSOCIATION NETWORKS

We apply the tracking-by-detection (TbD) paradigm, in which a tracker fuses object detections to generate object tracks that are consistent over time. [10], for example, provides a framework for analysing tracking approaches of tracking approaches that follow the TbD paradigm. We propose a tracker with a Long Short-Term Memory (LSTM) / bidirectional Long Short-Term Memory (BiLSTM) network for the prediction and association of the sensor objects (SO) to the existing tracks.

A. Single Prediction Network (SPENT)

Most existing tracking methods associate incoming detections pairwise with object states predicted by a simple motion model, e.g. a constant velocity model, using a Kalman filter. However, recent work has demonstrated that aggregating temporal information as well as contextual information can improve the tracking of multiple objects by utilising higher order information in addition to pairwise similarities between detections [4] [5] [11] [6]. Our approach is to dispense with the motion models and instead use the hidden states of the LSTM layer as an object-specific parameter set. The initial values of the hidden states of the LSTM layer are updated in tracking based on the measurement data received. This update therefore results in an internal correction over the course of the sequence.

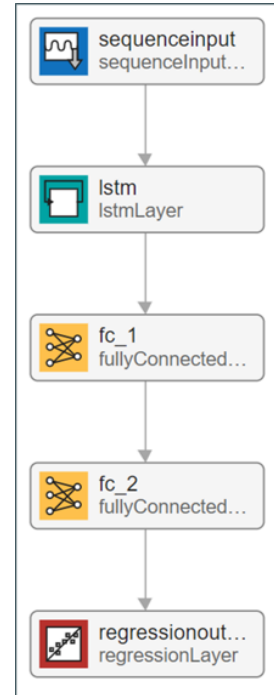


Fig. 2. Schematic representation of the generic network structure of SPENT.

a) *Data preprocessing:* Corresponding ground truth (GT) data from the KITTI data set was used for the development. A GT Track contains the information of an object over time, starting with its appearance and ending with its exit from the sensor detection range. In order to achieve better

generalisation and to increase the chance that the training converges, the track state values at time t (predictors) and track state values at time $t+1$ (targets) were normalised. After [12], normalisation can be performed so that the predictors and targets have a mean value of zero and a unit variance. The mean value μ and the standard deviation σ were calculated for all tracks using the following formulae:

$$\mu = \frac{1}{N} \sum_{i=1}^N A_i \quad (1)$$

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N |A_i - \mu|^2} \quad (2)$$

Pre-padding was also applied. [13] shows how padding influences the performance of neural networks in sequence-based tasks. The study suggests that although both pre-padding and post-padding are feasible, the choice of padding technique can have a significant impact on the efficiency of the model, especially for LSTM networks where the sequence context is crucial.

b) Network development: The Single Prediction Network (SPENT) was developed using the open-loop method, this means that the network predicts values for a future time step based on previously received data. For use in an online MOT process, the network is therefore able to make a prediction about the probable next state values using the state values received for an SO. The internal values (hidden states) of the LSTM layer are updated per time step based on the measurement data received. This updating process therefore provides an internal correction over the course of the sequence of each track.

The regression layer used (regressionoutput) calculates the loss of the mean square error for regression tasks. The mean square error (MSE) indicates the average of the squared difference between the model prediction and the target value. This value can be used as a measure of the quality of an estimator. For a single observation, the mean square error is given by:

$$MSE = \sum_{i=1}^R \frac{(t_i - y_i)^2}{R} \quad (3)$$

where R is the number of responses, t_i is the target output and y_i is the prediction of the network for sample i . For sequence-to-sequence regression networks such as SPENT, the loss function of the regression layer is half the mean square error of the predictions for each time step, normalised by S , not by R :

$$loss = \frac{1}{2S} \sum_{i=1}^S \sum_{j=1}^R (t_{ij} - y_{ij})^2 \quad (4)$$

where S is the sequence length. During training, the average loss is calculated using the observations in the mini-batch. Where S is the mini-batch length. In relation to the KITTI data set (cars and vans), an RMSE of 0.025 was achieved for the position prediction of all objects in the dataset.

B. Single Association Network (SANT)

SANT stellt einen datenbasierten Ansatz dar, welcher lernen kann, das kombinatorische Non Deterministic Polynomial Time (NP) hard Optimierungsproblem der Datenassoziation vollständig zu lösen. Im Vergleich zu den meisten Assoziationsverfahren [4] [5] [5] stellt SANT einen Ansatz dar, welcher als Inputdaten keine definierte Distanzmatrix erhält, sondern die bestehenden Tracks sowie jede neue Messung (SO) erhält. Damit entfällt die Definition eines Abstandsmaßes, womit dem Netzwerk die Freiheit gegeben wurde, einer eigenen Assoziationslogik zu folgen bzw. diese datenbasiert zu lernen. Somit ersetzt SANT die Berechnung eines Abstandsmaßes sowie einen Assoziationsalgorithmus wie z.B. den Hungarian Algorithm (HA).

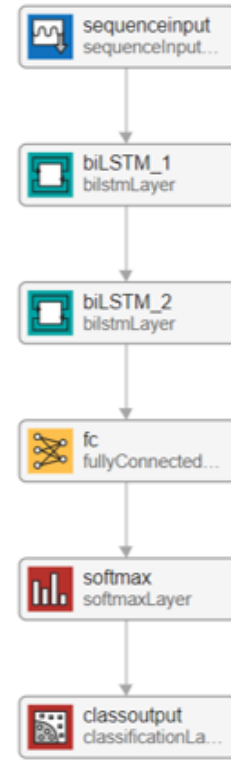


Fig. 3. Schematic representation of the generic network structure of SANT.

a) Datenvorverarbeitung: Für die Entwicklung des Single Association Networks (SANT) wurde die Datenassoziation als zeitlich gegliedertes Problem (Sequenz-zu-Vektor) betrachtet. Das Datenassoziationsproblem beschreibt somit die Zuordnung von einem SO zu einem Set von Tracks. Die entsprechenden Tracks wurden aus dem KITTI Datensatz der gelabelten Kameraaufzeichnungen extrahiert. Allerdings existieren für das betrachtete Problem der Assoziation keine realen GT Daten. Um sicherzustellen, dass die Zuordnung eines Sensor Objekts (SOs) zu einem Track eines Tracksets genau eine korrekte Lösung besitzt, wurde das SO in einem Zeitschritt aus dem bestehenden Trackset eines Zeitschritts erzeugt. Die Daten wurden entsprechend künstlich verrauscht, um aus den GT Daten realistische Sensordaten zu erzeugen. Das Datenformat wurde entsprechend gewählt, um eine in-

dexbasierte Trackzuordnung für das Single Association Network (SANT) zu ermöglichen. Die Größe der Inputmatrix entspricht daher $m * n$. Mit $m = 2 * \text{AnzahlZustandswerte}$ und $n = \text{Trackanzahl}(t)$.

b) *Netzwerkentwicklung*: Das Netzwerk (Fig. 3) ist als Sequenz-to-Classification ausgelegt, wobei der sequenceinput die beschriebene Inputmatrix pro Zeitschritt darstellt.

BILSTM,... Die vollständig verknüpfte Schicht (FC) gibt über die Anzahl der der Ausgabewerte die Klassenanzahl n vor. Die n Klassenwerte werden in der Softmax Schicht durch Anwendung der Softmaxfunktion in eine eindeutige Wahrscheinlichkeitsverteilung berechnet (wie in Kapitel 2.2.1 beschrieben). Durch Anwendung der Cross-Entropie-Operation in der Ausgabeschicht (Classout) konnte SANT auf einen korrekten Zuordnungswert (GT Klassenindex) trainiert werden.

Die Cross-Entropie-Kostenfunktion (Kreuzentropie) berechnet den Cross-Entropie-Verlust zwischen Netzvorhersagen und Zielwerten für die eindeutige Zuordnungsaufgabe (für sich gegenseitig ausschließende Klassen). Dabei wird mittels One-Hot-Kodierung die Klasse binär in einem Vektor dargestellt und somit ein 1-zu- n Code generiert. Nach folgender Formel werden die Crossentropie-Verlustwerte für jeden Eingabewert Y_j und zugehörigen Zielwert (Targetvalue) T_j elementweise berechnet:

$$\text{loss}_j = -(T_j \ln Y_j + (1 - T_j) \ln(1 - Y_j)) \quad (5)$$

Um einen Skalar loss zu erhalten werden alle Verlustwerte loss_j aufsummiert und durch die Anzahl der Samples N geteilt. Optional können die Verlustwerte von definierten Samples mit dem Gewichtungsfaktor w_j gewichtet werden:

$$\text{loss} = \frac{1}{N} \sum \text{loss}_j w_j \quad (6)$$

Eine entsprechende Nutzung des Gewichtungsfaktors w_j kann bei Datensätzen mit unausgewogenen (imbalanced) Klassenverteilung hilfreich sein.

C. Multi Association Network (MANTa)

- a) *Datenvorverarbeitung*:
- b) *Netzwerkentwicklung*:

IV. EXPERIMENTAL EVALUATION

... KITTI-Car Benchmark.

V. CONCLUSION

The conclusion goes here.

ACKNOWLEDGMENTS

This should be a simple paragraph before the References to thank those individuals and institutions who have supported your work on this article.

APPENDIX

PROOF OF THE ZONKLAR EQUATIONS

PROOF OF THE FIRST ZONKLAR EQUATION

Appendix goes here.

PROOF OF THE SECOND ZONKLAR EQUATION

And here.

REFERENCES

- [1] Jenny Seidenschwarz, Guillem Brasó, Victor Serrano, Ismail Elezi, Laura Leal-Taixé, "Simple cues lead to a strong multi-object tracker," *Paper*, 2022.
- [2] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, Ben Upcroft, "Simple online and realtime tracking," *Paper*, 2016.
- [3] Qi Chu, Wanli Ouyang, Hongsheng Li, Xiaogang Wang, Bin Liu, Nenghai Yu, "Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism," 2017.
- [4] Anton Milan, Seyed Rezaatofghi, Anthony Dick, Ian Reid, Konrad Schindler, "Online multi-target tracking using recurrent neural networks," 2016.
- [5] H. L. H. Z. C. Mertz, "Deepda: Lstm-based deep data association network for multi-targets tracking in clutter," 2019.
- [6] W.-C. H. H. K. T.-Y. L. Y. C. R. Yu, "Soda: Multi-object tracking with soft data association," 2020.
- [7] Q. C. W. O. H. L. X. W. B. L. N. Yu, "Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism," 2017.
- [8] T. M. A. K. L. L.-T. C. Feichtenhofer, "Trackformer: Multi-object tracking with transformers," 2022.
- [9] Roberto Henschel, Laura Leal-Taixé, Daniel Cremers, Bodo Rosenhahn, "Fusion of head and full-body detectors for multi-object tracking," *Paper*, 2017.
- [10] Ba-Tuong Vo, "code set for research use: Multi-sensor multi-target tracking," 2013. [Online]. Available: <https://ba-tuong.vo-au.com/codes.html>
- [11] Johannes Fitz, "Datenassoziation für multi-objekt-verfolgung mittels deep learning," *Paper*, 2020.
- [12] I. G. Y. B. A. Courville, *Deep Learning*. MIT Press, 2019.
- [13] D. M. R. N. V. S. Reddy, "Effect of padding on lstms and cnns," 2019.