Enhancing trajectory prediction of Simultaneous Collision Avoidance and Interaction modelling through parameter learning using Machine Learning

Soujanya Pradheepa Lohanathen
Department of Computer Science and Engineering
University of Moratuwa
Moratuwa, Sri Lanka
soujanya.18@cse.mrt.ac.lk

Chandana Gamage
Department of Computer Science and Engineering
University of Moratuwa
Moratuwa, Sri Lanka
chandag@cse.mrt.ac.lk

Sulochana Sooriyaarachchi
Department of Computer Science and Engineering
University of Moratuwa
Moratuwa, Sri Lanka
sulochanas@cse.mrt.ac.lk

***Abstract*—Autonomous driving is a hot topic throughout the world at present. When it comes to autonomous driving, tracking of road agents like vehicles and pedestrians is always an important issue to consider as it plays a vital role in trajectory prediction. Trajectories of road agents are dominated by various dynamic constraints. Simultaneous Collision Avoidance and Interaction Modelling (SimCAI) is a novel motion model for predicting the motion of road agents. It consists of techniques to avoid collisions between the road agents and to model safety interactions effectively. This research aimed to enhance trajectory prediction in traffic videos by optimizing parameter values and integrating the SimCAI model with a framework called TrackNPred for comparison purposes. Three machine learning search algorithms were employed to explore the parameter space and identify optimal configurations. Integration of SimCAI with TrackNPred allowed for evaluating and comparing its performance with Reciprocal Velocity Obstacles (RVO) in real-world scenarios. The research focused on improving SimCAI by learning the context-dependent parameter values through the search algorithms. Additionally, comparison techniques were employed to evaluate the accuracy of trajectory predictions. By identifying the best parameter values through the search algorithms, a reduction in Binary Cross-Entropy (BCE) loss was observed, indicating improved performance.**

***Index Terms*—Trajectory prediction, Road Agents, Reciprocal Velocity Obstacle, Heterogeneous environment**

I. Introduction

Trajectory prediction is the computation of future spatial coordinates of road agents such as vehicles and pedestrians. Trajectory prediction is critical in various fields such as autonomous vehicles, robotics, and human motion analysis. When given a history of states that a object has been for a time interval, trajectory prediction algorithms are able to predict the future states as in Figure 1. Several models are available for trajectory prediction and increasing their accuracy can improve safer and faster movements. For this, comparing and analyzing the state-of-art trajectory prediction models are essential. TrackNPred [1] is an existing software framework that visualizes and compares some of the existing trajectory prediction models. Road Track is a trajectory prediction model presented by Chandra et al [2] for predicting the spatial coordinates of road agents. This is achieved by modeling collision avoidance and interactions between road agents for the next frame in a video stream.

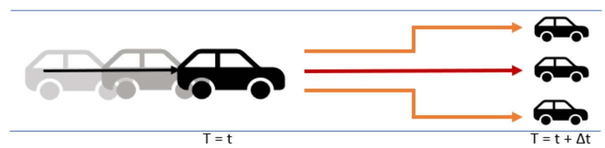

Fig. 1: Explanatory image of trajectory prediction

This paper is structured as follows. Section I introduces the research and describes the objectives. Section II describes the key frameworks that underpin the research work presented in this paper and the related work in the fields of trajectory prediction, motion models, and search algorithms. Section III explains the methodology of the research. Section IV presents the evaluation methods carried out and discusses the results followed by the conclusions in section V.

II. Literature Review

TrackNPred [1] and Road Track [2] are the main frameworks that have been improved by this research. SimCAI model introduced in Road Track and TrackNPred is developed as a framework to evaluate and compare the already existing deep learning techniques in the field of trajectory prediction.

A. Road Track

Road Track [2] is an end-to-end Trajectory Prediction scheme. The Road Track algorithm addresses the problem of tracking road agents in videos. The goal is to assign a unique ID to each road agent in every frame of a given video. To accomplish this, the algorithm combines two key components: Mask R-CNN for object segmentation and SimCAI for predicting the next state of the road agents.

Initially, Mask R-CNN is employed to perform pixel-wise segmentation of the road agents in the video. This process generates a set of segmented boxes that represent the detected road agents. From these boxes, the algorithm extracts relevant features using a deep learning-based feature extraction architecture fine-tuned specifically on traffic data sets. This enables the algorithm to learn meaningful features associated with traffic patterns.

SimCAI plays a crucial role in predicting the next state of each road agent for the subsequent frame. The state consists of spatial coordinates and velocities. By leveraging SimCAI, the algorithm generates another set of segmented boxes for each road agent at the next time step $(t+1)$.

To assign IDs to road agents in the next frame, the algorithm utilizes a Convolutional Neural Network (CNN) to compute features from the segmented boxes. These features are then compared using association algorithms that incorporate the Cosine metric and the IoU overlap metric.

This metric constructs a cost matrix Σ to quantify the degree of overlap between each predicted bounding box and all nearby detection bounding box candidates. $\Sigma(i,j)$ stores the IoU overlap between the bounding box of p_i and that of h_j and is calculated as follows where H_i is the subset of all detected road-agents in the current frame that are within a circular region around agent p_i :

$$\Sigma(i,j) = \frac{B_{p_i} \cap B_{h_j}}{B_{p_i} \cup B_{h_j}}, \text{ where } h_j \in H_i.$$

If the cosine metric id is denoted as C and the Intersection over Union (IoU) overlap metric is denoted as I, then the combined cost function value is obtained as follows:

Combined Cost = $\lambda_1 C + \lambda_2 I$, with $\lambda_1 + \lambda_2 = 1$,

where λ_1 and λ_2 are constants representing the weights for the individual metric costs. Matching a detection to a predicted measurement with the highest overlap becomes a maximum-weight matching problem, which we efficiently solve using the Hungarian algorithm [3]. The ID of the road-agent at time t is assigned to the road-agent at time $t+1$ whose appearance is most closely associated with the road-agent at time t. By combining object segmentation, feature extraction, prediction, and association techniques, the Road Track algorithm provides an effective solution for tracking road agents in videos.

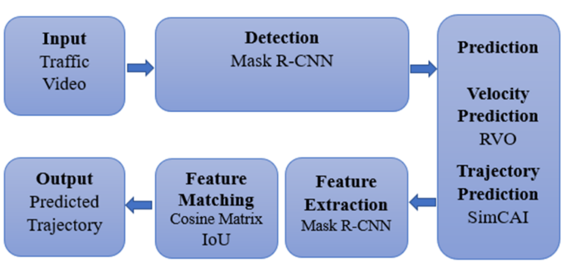

Fig. 2: Overview of Road Track

B. TrackNPred

TrackNPred [2] is a Python-based software (See Figure 3) designed for real-time trajectory prediction for autonomous road agents. The primary objective of TrackNPred is to facilitate the safe navigation of autonomous road agents in dense and diverse traffic scenarios by estimating the movements of nearby road agents in the upcoming few seconds.

A key goal of TrackNPred is to provide researchers with a comprehensive deep-learning tool that enables trajectory prediction based on various state-of-the-art neural network architectures. These architectures include Generative Adversarial Networks (GANs), Recurrent Neural Networks (RNNs such as LSTMs), and CNNs. Researchers can select the desired network architecture and modify individual components without disrupting the overall framework.

TrackNPred integrates real-time tracking algorithms with end-to-end trajectory prediction methods, creating a robust framework. The input to TrackNPred is a video captured by a moving or static RGB camera. The system employs a tracking module to generate trajectories for each road agent by selecting an appropriate tracking method. These trajectories serve as the history for each agent in the trajectory prediction module. The final output of TrackNPred is the predicted future trajectory for the road agent within a specific prediction window.

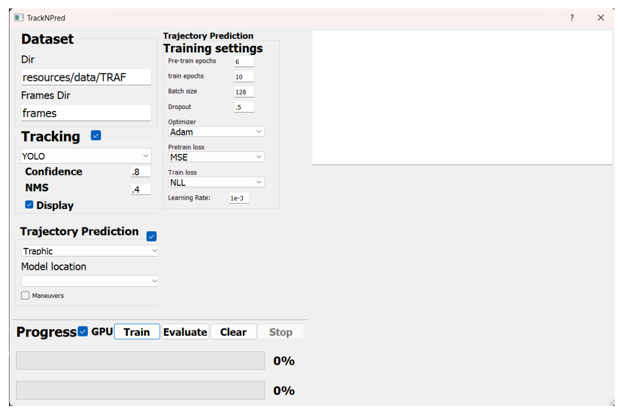

Fig. 3: Running TrackNPred System

A significant distinction of TrackNPred from existing trajectory prediction methods in the literature is its independence from manually annotated input trajectories. Unlike other approaches that rely on ground truth trajectories, TrackNPred does not require such annotations, making it more versatile and applicable to various scenarios.

Additionally, TrackNPred includes an evaluation and benchmarking component that assesses the real-time performance of different trajectory prediction methods on a real-world traffic dataset. This dataset consists of over 50 videos featuring dense and heterogeneous traffic, including various types of road agents such as cars, buses, trucks, pedestrians, scooters, motorcycles, and others. The dataset encompasses different camera viewpoints, motion conditions, times of day, and density levels, providing a comprehensive evaluation platform.

C. Human Motion Predictions

The social force model is presented as a powerful approach for realistic human motion predictions. It considers interactions between pedestrians incorporating aspects such as motivation, repulsion, and physical constraints [4]. By combining a people tracker with the social force model, more accurate and realistic human motion predictions can be achieved. The social force model is simple and capable of handling inter-people relations, making it a promising approach for human motion prediction.

D. People and Vehicle Tracking

Multiple studies discuss the importance and challenges of reliable detection and tracking of people and vehicles in various scenarios. Laser-based systems combined with visual information have been proposed for real-time monitoring and tracking of multiple people [5]. The fusion of laser-based and vision-based tracking using a Bayesian formulation demonstrates effectiveness in tracking multiple people simultaneously.

In crowded video scenes, leveraging structured patterns of crowd motion using Hidden Markov Models (HMMs) has shown improved tracking results [6]. Sparse and noisy sensor data collected by sensors can be used for estimating peoples' locations, and a particle filter combined with Voronoi graphs has been proposed for accurate tracking [7]. Hybrid motion models combining discrete and continuum models have been introduced for pedestrian tracking in medium to high-density crowd videos [8].

E. Autonomous Vehicles and Traffic Control

The development of autonomous vehicles relies on scene understanding and short-term prediction for reliable navigation in challenging environments [9]. Detailed characterization of the navigable space is essential for safe navigation. Traffic congestion and efficient use of existing infrastructure are critical issues that require better traffic information and control systems [10]. Vehicle tracking via video image processing has been proposed as a promising approach to provide traditional traffic parameters as well as new parameters such as lane changes and vehicle trajectories [11].

F. Motion Models for Tracking

Weak assumptions about human motion, such as the Brownian model or constant velocity motion model, have limitations in representing complex, nonlinear human motion patterns [12]. Various improved motion models have been proposed, including learning goal locations, extracting Voronoi graphs, and deriving motion patterns from trajectories. The social force model, which considers interactions between pedestrians, has shown promising results in realistic human motion prediction.

G. Search Algorithms in Parameter Tuning

Parameter tuning is crucial for optimizing the performance of evolutionary algorithms (EA) and deep neural networks. The lack of widely used parameter-tuning algorithms in evolutionary computing has been identified [12]. Automated hyper-parameter optimization (HPO) techniques, such as random search and genetic algorithms, have been explored to address this challenge [13], [14]. Comparative studies have evaluated different HPO methods, including Bayesian optimization, to improve model accuracy and stability [15].

Overall, the currently existing SimCAI model is the most effective trajectory prediction algorithm as per the authors [2]. However, the available implementation of SimCAI supports only a simulated environment that is manually defined (Figure 4).

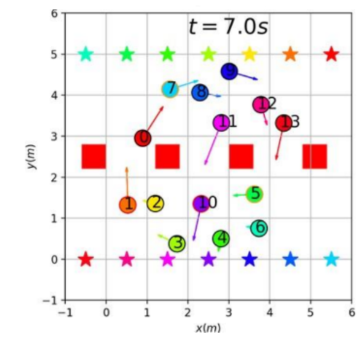

Fig. 4: Currently Existing Simulated Environment of SimCAI

III. Methodology

Based on the literature review findings, three search algorithms have been selected for testing parameter tuning: Grid search, Random search, and Bayesian search. These algorithms offer different approaches to exploring the parameter space and identifying optimal configurations. By evaluating the performance and efficiency of each algorithm, valuable insights can be gained regarding their suitability for enhancing SimCAI.

Additionally, it has been observed that existing implementations of SimCAI and RVO (Reciprocal Velocity Obstacles) only support simulated environments. To overcome this limitation, a decision has been made to modify and integrate these models with the TrackNPred framework. This integration will facilitate the evaluation and comparison of SimCAI with RVO in real-world scenarios, providing a comprehensive understanding of their respective capabilities and limitations.

The research aims to enhance the trajectory prediction capabilities of SimCAI by optimizing its parameter values through the utilization of machine learning search algorithms. SimCAI, as identified in the literature review, is currently regarded as an outperforming model for trajectory prediction. However, its performance can be further improved by selecting the most suitable parameter values, which are typically chosen heuristically.

The research employs machine learning search algorithms and integrate SimCAI with the TrackNPred framework. This integration allows for the evaluation of SimCAI's performance in real-world environments, expanding its practical applicability. By conducting experiments and assessments within real-world scenarios, a more comprehensive understanding of SimCAI's effectiveness and limitations can be obtained. The methodology section of the research paper provides detailed explanations of the steps taken to carry out the study.

As shown in Figure 5, methodology for the system being developed involves several steps to improve trajectory prediction in traffic videos. The input for the system is a traffic video, which is processed using the Mask R-CNN (Region-based Convolutional Neural Network) for object detection. This allows the system to identify and track vehicles in the video.

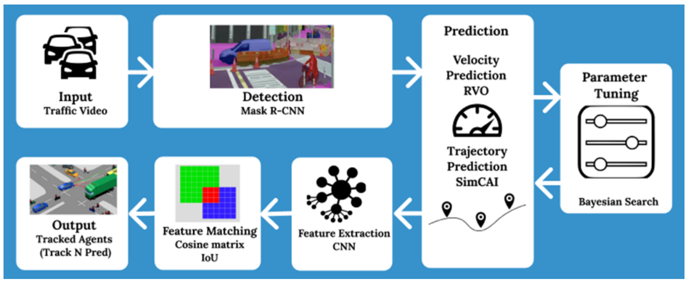

Fig. 5: Architecture of the methodology

For trajectory prediction, the existing Road track algorithm is utilized. Within this algorithm, the RVO approach is employed for velocity prediction, and SimCAI is utilized for trajectory prediction.

To evaluate the accuracy of the trajectory predictions, the features of the predicted images are extracted using another Mask R-CNN model. These extracted features are then matched with the features of the original image using cosine similarity matrix and IoU techniques. This comparison allows for the assessment of the accuracy and similarity between the predicted and actual trajectories.

Finally, the output of the system, which includes the predicted trajectories and their evaluation, is displayed within the integrated TrackNPred environment. This environment provides a comprehensive platform for visualizing and analyzing trajectory predictions in real-world scenarios.

Overall, the methodology encompasses the steps of object detection, velocity, and trajectory prediction using RVO and SimCAI, parameter optimization using machine learning search algorithms, feature extraction, trajectory evaluation, and visualization within the TrackNPred environment. This systematic approach aims to improve the accuracy and applicability of trajectory prediction in traffic videos.

A. Integrating SimCAI and TrackNPred

Since the implementation of the Road Track algorithm is not available in a public repository, there is a need to develop a parent algorithm that consists of the implementations of RVO and SimCAI. It also needs to deal with the real-world data. For the parent algorithm, TrackNPred was chosen since it already supports real data and it also can evaluate and compare the trajectory prediction models. According to the literature review carried out, it was the latest framework that has the above-mentioned features. Therefore, it is believed that using TrackNPred in place of Road Track is a better option. However, since the currently existing implementations for RVO and SimCAI only support simulated environments, it is not possible to integrate the implementations with TrackNPred as they currently exist.

In order to deal with the real data and for the integration, there needs to be a Neural Network model which can be called whenever the prediction needs to be done. Therefore, a neural network model which consists of the modified implementations of SimCAI and RVO has been developed to support this requirement.

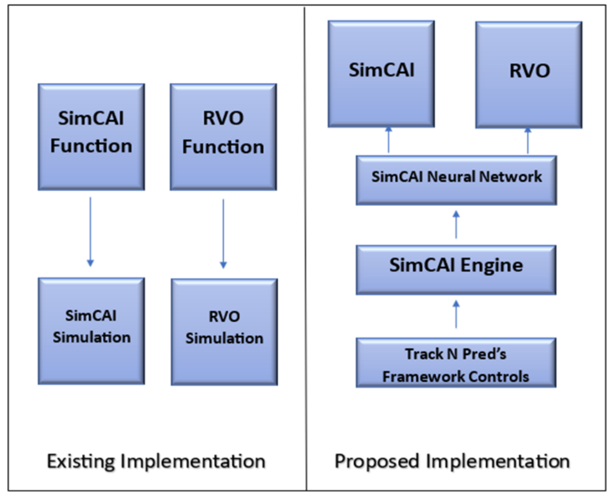

Fig. 6: Comparison between the existing and proposed system of Road Track

As shown in Figure 6, the existing implementation consists of simulations for each component. The proposed system consists of the newly developed SimCAI neural network which consists of both RVO and SimCAI. Then a SimCAI Engine is designed to be called from TrackNPred Environment.

B. Improving SimCAI model by Parameter Learning

As stated previously, SimCAI depends on some parameter values (See Figure 7) which are heuristically defined in existing implementations. These values play a vital role in determining the accuracy of the trajectory prediction.

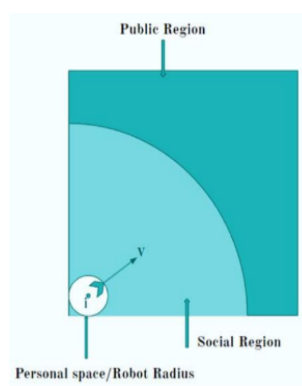

Fig. 7: SimCAI's parameters

To select the best parameter values, the three best search algorithms as found through literature survey were considered. It is proposed to implement all three algorithms to choose the best fitting values for the context-dependent parameters and choose the best searching algorithm among them. To achieve this efficiently, the source code of these three algorithms has been revamped and used in the modified neural network model of SimCAI. The context parameters were listed within an array and each search algorithm has been configured to train the neural network. To incorporate these changes the SimCAI and RVO algorithms have been modified to map the real-world data into the simulation environment. For this purpose, the result from the Mask R-CNN segmentation has been used as the input positions for SimCAI. The Goal position for each agent is extracted from the last frame detected. Then the current velocity for each road agent was assumed to be constant over the detected frames and it was computed using the displacements in x and y directions across the detected frames. Finally, these data altogether have been fed into the SimCAI simulation and the output results are observed.

IV. Testing & Results

The evaluation for this research has been conducted in two ways to determine how well the two objectives are achieved.

A. Testing the Integration of SimCAI and TrackNPred

SimCAI has been developed as a neural network to be integrated with TrackNPred. Once integrated, the successful integration is evaluated by performing all the features that exist in the TrackNPred system using SimCAI. The observation showed that all the features have been working well with the SimCAI model. Also, a UI has been developed to integrate the SimCAI model changing option in the Trajectory Prediction section (See Figure 8).

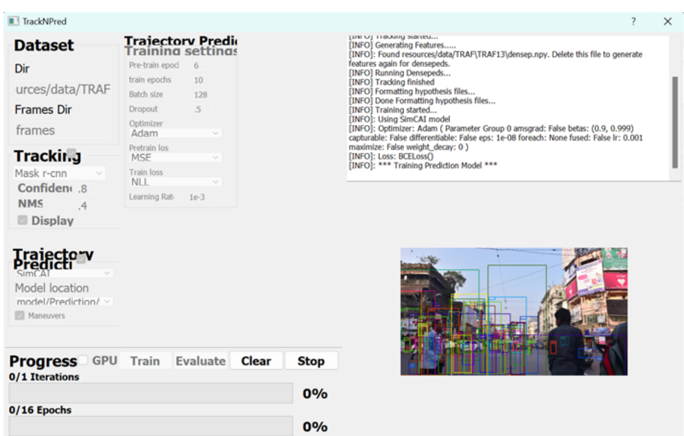

Fig. 8: Integration of SimCAI and TrackNPred

B. Testing the Identified Parameter Values

To evaluate the performance of the model after identifying the best values for the parameters, a comparison test has been carried out. Initially, the SimCAI neural network model is tested with the default values for the parameters that are assigned in the existing SimCAI implementation. Then the model has been run with the predicted values for the parameters from the search algorithms. While running the tests, the best parameters identified for the social region, steering angle, personal space, and robot radius are 0.1, 30, 0.01 and 1 respectively. It has been observed that the BCE loss is reduced when using the suggested values for the context-dependent parameters as in Table I.

Epoch	SimCAI (BCE loss)	Modified SimCAI (BCE loss)
1	5.43	5.39
2	5.45	5.29
3	5.34	5.30
4	5.46	5.21
5	5.45	5.26

TABLE I: Comparison of BCE loss between existing and modified models

In addition to integrating SimCAI and TrackNPred as well as modifying SimCAI using parameter learning, the existing SimCAI implementation was modified to support real data. It has been developed to get the detections from the Mask R-CNN output. Then it will extract the goal position for each road agent and compute the velocities for each road agent using the displacements in the x and y directions across the frames. It has been also modified to mark the real-world road agents in the simulated environment and mark the trajectories. For instance, the real data interpreted in the SimCAI environment for a particular video at time $t = 0$ is shown in Figure 9 and the data at $t = 15s$ is interpreted as in Figure 10.

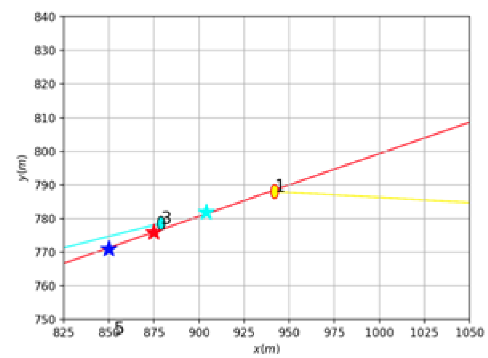

Fig. 9: Real Data in SimCAI environment at t=0

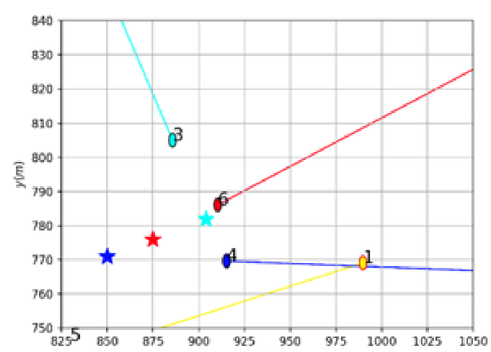

Fig. 10: Real Data in SimCAI environment at t=15s

V. Conclusion

In conclusion, this research aimed to enhance the trajectory prediction capabilities of SimCAI by optimizing its parameter values through the utilization of machine learning search algorithms. The selected search algorithms, namely Grid search, Random search, and Bayesian search, provided different approaches to exploring the parameter space and identifying optimal configurations.

Additionally, the research addressed the limitation of existing implementations of SimCAI and RVO, which only supported simulated environments, by integrating them with the TrackNPred framework to evaluate their performance in real-world scenarios.

The methodology involved several steps, including object detection using Mask R-CNN, velocity, and trajectory prediction using RVO and SimCAI, parameter optimization using machine learning search algorithms, feature extraction, trajectory evaluation, and visualization within the TrackNPred environment. The integration of SimCAI with TrackNPred was achieved by developing a neural network model that combined the modified implementations of SimCAI and RVO. This integration allowed for the evaluation and comparison of trajectory prediction models in real-world scenarios.

The testing and results section focused on evaluating the integration of SimCAI and TrackNPred, as well as testing the identified parameter values. The successful integration of SimCAI with TrackNPred was confirmed through the evaluation of all features within the TrackNPred system using SimCAI. The comparison test between the default parameter values and the values identified by the search algorithms showed a reduction in the BCE loss when using the suggested context-dependent parameter values.

Overall, this research contributes to improving the accuracy and applicability of trajectory prediction in traffic videos. By optimizing parameter values and integrating SimCAI with the TrackNPred framework, the research provides valuable insights into enhancing trajectory prediction capabilities of SimCAI. These findings pave the way for further advancements in trajectory prediction models and their practical implementation.

References

[1] R. Chandra, U. Bhattacharya, T. Randhavane, A. Bera, and D. Manocha, "RoadTrack: Realtime Tracking of Road Agents in Dense and Heterogeneous Environments," Proc. - IEEE Int. Conf. Robot. Autom., no. May, pp. 1270–1277, 2020, doi: 10.1109/ICRA40945.2020.9196612.

[2] R. Chandra, U. Bhattacharya, C. Roncal, A. Bera, and D. Manocha, "ROBUSTTP: End-to-end trajectory prediction for heterogeneous road-agents in dense traffic with noisy sensor inputs," ACM Computer Science in Cars Symposium, 2019.

[3] Harold W Kuhn. The hungarian method for the assignment problem. In 50 Years of Integer Programming 1958-2008, pages 29–47. Springer, 2010.

[4] Jinshi Cui, Hongbin Zha, Huijing Zhao, and R. Shibasaki, "Tracking multiple people using laser and Vision," 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005. doi:10.1109/iros.2005.1545159

[5] M. Fernández-Sanjurjo, B. Bosquet, M. Mucientes, and V. M. Brea, "Real-time visual detection and tracking system for traffic monitoring," Engineering Applications of Artificial Intelligence, vol. 85, pp. 410–420, 2019. doi:10.1016/j.engappai.2019.07.005

[6] X. Zhao, D. Gong, and G. Medioni, "Tracking using motion patterns for very crowded scenes," Computer Vision – ECCV 2012, pp. 315–328, 2012. doi:10.1007/978-3-642-33709-3_23

[7] Lin Liao, D. Fox, J. Hightower, H. Kautz, and D. Schulz, "Voronoi tracking: Location estimation using sparse AND NOISY SENSOR DATA," Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453). doi:10.1109/iros.2003.1250715

[8] A. Bera and D. Manocha, "Reach - realtime crowd tracking using a hybrid motion model," 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015. doi:10.1109/icra.2015.7139261

[9] H. A. Ignatious, H. El-Sayed, M. A. Khan, and B. M. Mokhtar, "Analyzing factors influencing situation awareness in autonomous vehicles—A survey," Sensors, vol. 23, no. 8, p. 4075, 2023. doi:10.3390/s23084075

[10] M. Haris and J. Hou, "Obstacle detection and safely navigate the autonomous vehicle from unexpected obstacles on the driving lane," Sensors, vol. 20, no. 17, p. 4719, 2020. doi:10.3390/s20174719

[11] B. Coifman, D. Beymer, P. McLauchlan, and J. Malik, "A real-time computer vision system for vehicle tracking and traffic surveillance," Transportation Research Part C: Emerging Technologies, vol. 6, no. 4, pp. 271–288, 1998. doi:10.1016/s0968-090x(98)00019-9

[12] D. Kelly and F. Boland, "Motion model selection in tracking humans," IET Irish Signals and Systems Conference (ISSC 2006), 2006. doi:10.1049/cp:20060464

[13] A. E. Eiben and S. K. Smit, "Parameter tuning for configuring and analyzing evolutionary algorithms," Swarm and Evolutionary Computation, vol. 1, no. 1, pp. 19–31, 2011. doi:10.1016/j.swevo.2011.02.001

[14] L. Villalobos-Arias and C. Quesada-López, "Comparative study of random search hyper-parameter tuning for software effort estimation," Proceedings of the 17th International Conference on Predictive Models and Data Analytics in Software Engineering, 2021. doi:10.1145/3475960.3475986

[15] A. M. Vincent and P. Jidesh, "An improved hyperparameter optimization framework for AUTOML systems using evolutionary algorithms," Scientific Reports, vol. 13, no. 1, 2023. doi:10.1038/s41598-023-32027-3

[16] M. Feurer and F. Hutter, "Hyperparameter optimization," Automated Machine Learning, pp. 3–33, 2019. doi:10.1007/978-3-030-05318_1